National Defence / Défense nationale

Canada

LAYERED PROTOCOL ANALYSIS OF A CONTROL DISPLAY UNIT

Defence and Civil
INSTITUTE OF ENVIRONMENTAL MEDICINE

INSTITUT DE MEDECINE ENVIRONNEMENTALE
pour la défense

1133 Sheppard Avenue West, PO Box 2000, North York, Ontario, Canada M3M 3B9
Tel. (416) 635-2000 Fax. (416) 635-2104

EXECUTIVE SUMMARY

LAYERED PROTOCOL ANALYSIS OF A CONTROL DISPLAY UNIT

Philip S. E. Farrell and Marc A. H. Semprie,
Defence and Civil Institute of Environmental Medicine, North York, Ontario, CANADA

Humans and machines interact through the controls and displays of an interface. The interface should provide the appropriate feedback to both the human and the machine in order to reduce the risk of misinterpretation, and increase system performance.

In aircraft environments, crews often seem to have moments of high workload, resulting in poor decision making and errors. Computer-based electronic equipment, such as the Control Display Unit (CDU), have been added to the suite of cockpit instruments in an attempt to reduce workload. However, in some cases, the new *glass cockpit* technologies have exacerbated the workload problem.

Land Aviation Test and Evaluation Facility (LATEF) has recognised several areas of high workload due to the CDU operation in the CH-146 helicopter related to the design of the CDU interface. This paper uses Layered Protocol Theory (LPT) to analyse the interaction between the pilot and the CDU, leading to interface design requirements. Aspects of the interface were identified and re-designed in order to ensure proper and timely feedback.

LPT has been described as a special case of Perceptual Control Theory (PCT). It states that, *all communication is the control of belief*. LPT introduces a general framework that describes all types of communication. The framework involves interpreting and comparing current and desired belief states within both partners, and using any discrepancy to design and transmit appropriate messages. The theory is hierarchical, recognising that communication takes place at many abstract levels simultaneously. For example, a thought might be constructed with sentences, and sentences with words, words with letters, letters with shapes, etc.

The interaction model began with the desired belief state: *to believe a radio link has been set*. The model ended with the lowest abstract level of the required keys and displayed information. The analysis identified interaction deficiencies observed by LATEF as well as other potential problems that may arise. A proposed interface was designed that addressed all the deficiencies. The next step is to compare the proposed and current interfaces in a test environment, and record any improvements in performance or workload.

ABSTRACT

Layered Protocol Theory (LPT) has been described as a special case of Perceptual Control Theory (PCT) where its core tenet is, *All communication is the control of belief.* It was recognised that LPT could be used to analyse the interaction between communicating partners in the context of human-machine systems. System interface problems were identified for the Control Display Unit (CDU) in the CH-146 Griffon helicopter. This application presented a good opportunity to conduct a Layered Protocol analysis on the pilot-CDU system. Aspects of LPT were discussed in detail including the LPTool, its Network View, GPG View, and Nine Element View. A pilot-CDU interaction was modelled with the aid of the LPTool program. The analysis yielded a list of interaction deficiencies between pilot and CDU which supported previous observations. The deficiencies were addressed in a new interface design that would provide the necessary controls and displays so that the required messages could be successfully transmitted and interpreted.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

On December 20, 1995, at about 2138 e.s.t, American Airlines, Flight 965, a regularly scheduled passenger flight from Miami, FL to Cali, Colombia, crashed 38 miles north of Cali into mountainous terrain during a descent under instrument flight rules. There were 156 passengers and 8 crewmembers aboard. Four passengers survived the accident. Eight Colombians died during the rescue attempt.

The crew had fallen behind monitoring the flight progress as they commenced their descent to the Cali airport. While trying to fly a revised clearance the crew became geographically disoriented. In attempting to recover from this situation they tried to select the ROZO Non-Directional Beacon (NDB) and fly directly to this radio aid rather than locate, and return to, the initial approach at the TULUA Visual Omni-Range (VOR).

The onboard Flight Management System (FMS) aids in waypoint selection by displaying a list of NDB identifiers. All identifiers beginning with the letter *R*, in this case, would be displayed in order of airport size. The ROMEO NDB near Bogota appeared first on the list. However, the ROZO NDB did not appear anywhere on the list since the full four letter code, *ROZO*, was required to distinguish between the ROMEO and ROZO identifiers which had the same frequency and morse identifier, *R*.

The flight crew made a logical assumption that the ROZO NDB was closest to the aircraft and would be first on the list. By entering the letter *R*, and selecting the first NDB, the FMS initiated a left hand turn back towards Bogota and into high terrain before the crew could recognise and correct the error.

Communication and Navigation are critical for flight. In carrying out these tasks the pilot must consider various factors such as radios, security, frequencies, waypoints, communication modes, etc. The Control Display Unit (CDU) provides the interface between the pilot and the aircraft avionics system in monitoring the aircraft systems and mission data. This includes establishing radio links and setting waypoints. Interaction between the crew and the FMS (including the CDU), are not always intuitive and must be made explicit under most circumstances, as evident by the Cali example.

During the pilot-CDU interaction, the pilot must clearly perceive what the system is doing in order to make appropriate decisions that move the system closer to some desired state. As the CDU provides more options for the pilot to manipulate the aircraft path, it becomes increasingly more difficult for the pilot to monitor all the goals, actions, and reactions within this complex system. The FMS system should assist the pilot in monitoring critical goals and comparing them with their current and predicted states. Any significant deviation from the goals might be flagged.

The problems experienced in commercial glass cockpits have been identified in their military counterparts. In 1995, Land Aviation Test and Evaluation Flight (LATEF) Operation Liaison and Acceptance (OLA) section recognised that new technologies and operating procedures run the risk of imposing more workload on the CH-146 Griffon flight crew if not properly implemented. Soon after, the CDU was identified as contributing significantly to the workload of the crew.

Human-machine analysis techniques can be applied to this problem in order to minimise workload and increase system performance. Traditional methods of human-machine analysis treat the human as a single component in the system (Sinaiko and Buckley, 1957), transforming sensory information into actions. The goals and intentions for the system are usually defined external to the human or machine. Traditional methods tend to deal with one level of abstraction at a time, usually the interface level (e.g., making sure that a certain button press displays the expected list of NDB frequencies under all mission conditions).

Perceptual Control Theory (PCT) (Powers, 1973) is an alternative view of human-machine interaction where the human model is expanded to include many levels of abstraction. Powers includes the goals, perceptions, and decision making as part of the human component, but the machine and its environment is modelled as a single component, transforming behaviours and actions into sensory information for the human to process. PCT describes how higher level goals (e.g., *to stay on course*) are distributed amongst the lower level control loops which have specific subgoals (e.g., *to enter the appropriate identifier letter*). The model can be described down to the interface level where traditional methods of human-machine interaction may be invoked.

Layered Protocol Theory (LPT) (Taylor, 1993) claims to provide another technique for the analysis of human-machine interaction by treating the machine as a communicating partner in simple conversation with the human. The communication model involves hierarchical structures for each partner. Each level of the hierarchy is represented by the control of common beliefs and perceptions. At the interface level, the actions and sensory information are passed between the two hierarchical structures.

The concept of Virtual messages between the two structures is unique to LPT. A Virtual message provides feedback to the partners relative to a particular level of abstraction. It is the combination of all the lower level messages. Human-human interaction involves the conscious transmission of Virtual messages such as thoughts, sentences, or words, as well as the unconscious, skill-based, or automatic transmission of *real messages* (i.e., actions and sensory information) such as producing vocal vibrations or drawing lines and circles that produce letters on paper.

The intention of applying LPT to human-machine interaction is not to suggest that machines are *intelligent* or self aware. Neither is the intention to suggest that machines can be designed to have perceptions or beliefs. More precisely, machines reflect the beliefs and intentions of their designers. However, due to the nature of the LPT analysis, it is convenient to attribute human-like concepts and expressions to the machine. Once the model is analysed, the results still must be translated into design specifications for the machine.

From an LPT perspective, the pilot and CDU may be seen as communicating partners, each wanting to move their current beliefs towards desired beliefs about themselves and each other. Virtual messages are used to change beliefs. An analyst may determine interaction deficiencies by modelling the communication and mapping out potential belief states and feedback messages that would be encountered for a given communiqué.

This paper reports on a pilot-CDU interaction model using Layered Protocol Theory. The first sections provide some background for LPT. Next, the software program used to analyse the pilot-CDU interaction, LPTool, is introduced. The analysis of the pilot-CDU interaction is described in detail, followed by a discussion of the results.

2. PERCEPTUAL CONTROL AND LAYERED PROTOCOL THEORIES

Layered Protocol Theory is an extension of Perceptual Control Theory (PCT) (Powers, 1973) which describes the human information processing system in terms of Classical Control Theory (Van de Vegte, 1990). That is, as humans interact with their environment, information is being collected and decisions are being made so that perceptions of environmental variables move towards internal reference or goal perceptions. In this context, the perception is the control variable, and Classical Control Theory provides powerful techniques for this type of system analysis.

2.1 Perceptual Control Theory

Traditionally, psychology has adopted the belief that behaviours are being controlled in response to sensory information. In PCT terms, however, the sensory information comes from disturbances and behaviours acting on the world, from which perceptions are generated and compared to goal perceptions. A person then acts on the world (or behaves) so that the perceptions are driven towards their goal states. Thus, behaviour is a result of the control of perception.

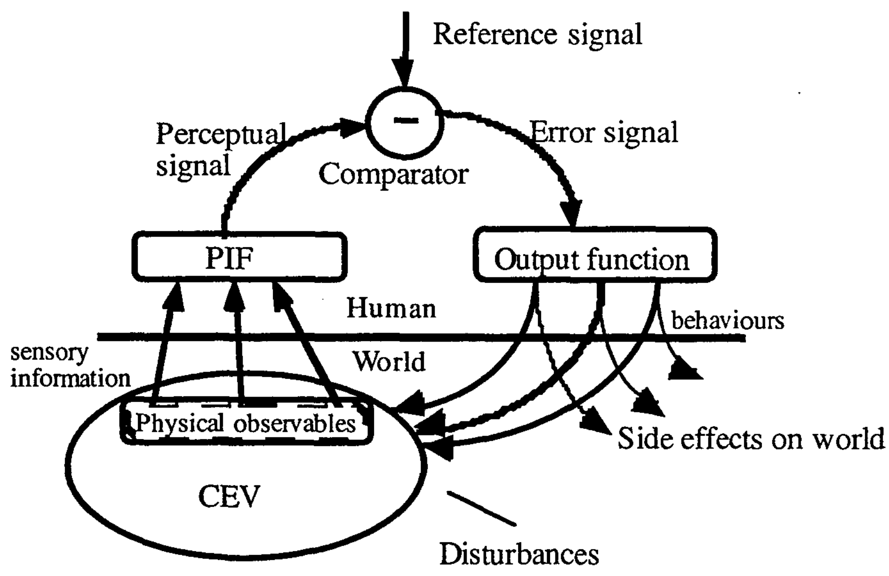


Figure 1. A closed loop control unit, or Elementary Control Unit in PCT.

The structure of PCT is a closed feedback loop called an *Elementary Control Unit* (ECU) as shown in Figure 1. The *Complex Environmental Variable* (CEV) represents a collection of physical observables in the world that is related to a particular perception for the given ECU. The CEV is potentially visible to an outside observer. However, the perception of the CEV may differ from person to person. For example, in recounting a car accident, it is rare that two people would make identical eye witness reports although there was a single set of sensory information available to each of them. The

difference is due to how individuals interpret the information as well as how they design a response.

The *Perceptual Input Function* (PIF) processes the sensory information originating from CEV and transforms it into a *Perceptual* signal. The PIF is part of the internal mental process and cognitive structure. It involves levels of processing, strategies, mental models, etc. (Hendy, 1994) in order to generate the Perceptual signal. PCT does not provide the exact details about the nature of this transformation, but system identification techniques may yield a model that closely matches the observed transformations.

The *Comparator* compares the Perceptual signal and the *Reference* signal resulting in an *Error* signal. A non-zero Error signal means that the current perception does not match its goal state and the error becomes the impetus to act on the world (behaviour), or generate lower level Reference signals. This implies that a single ECU is usually part of a hierarchy of ECUs. That is, a single perception may be the result of the combination of several lower level perceptions, all of which are controlled.

The Error signal is operated on by the *Output function.* Like the PIF, the Output function takes into consideration the person's mental model before producing a behaviour. Thus, the Output function is difficult to describe deterministically due to its variability amongst people. The error and Output function operation results in behaviours that act on the world. The world, in turn, produces sensory information and closes the ECU loop.

An ECU is stable when the Error signal tends to zero or, at least, is bounded. However, *Disturbances* may excite the system leading to a departure of the Perceptual signal from its set point. Classical Control Theory states that control algorithms can be implemented such that the Reference signal is achievable despite the disturbance. Such algorithms are proposed to be embedded in the PIF and Output function, and operate only when the Error signal value falls within a bounded region of the system's state space. An error outside of this region may become unstable resulting in either a change in strategy at the local level or a re-organisation of ECUs within the proposed hierarchical structure of a PCT model. The global re-organisation has been coined a *bomb* (Taylor, 1993).

Side Effects are behaviours that effect other CEVs. For instance, if two patrons are on an elevator and one wants to go up and the other down, then a single patron's behaviours will simultaneously influence both perceptions. One ECU will be locally moving away from the set point while the other is settling. In a cooperative dialogue, one can imagine Side Effects being mutually beneficial to both ECUs. For instance, one of the patron's may adjust their lower level goals to achieve the desired perception (e.g,, take the stairs). Thus the ECUs will be stabilised simultaneously.

2.2 Layered Protocol Theory

Layered Protocol Theory (LPT) originated from studying language and communication. Taylor (1993) recognised that LPT provided a framework where the interaction between two communicating partners may be analysed. The counterpart to the ECU is called a Protocol Node (PN). Figure 2 shows the related elements between the PN and ECU. The fundamental difference between LPT and PCT is that while the ECU represents the control of one perceptual signal, the protocol node (PN) represents the control of a vector of perceptions that form a single belief.

The Primal message is a desired belief state that a partner (usually, the originator) wants to be in. The Coder and Decoder designs and interprets, respectively, messages so to communicate the Primal message. Inherent in the framework is the necessity for feedback so that one partner can evaluate the current belief state of the other partner. The arrows in and out of the Decoder and Coder in Figure 2 are Feedback messages that filter down within lower level protocol nodes, through the interface, and up into the partner's protocol hierarchy. Two PNs connected in a loop represent two communicating partners as depicted in Figure 3. The connecting arrows are virtual Feedback messages that represent all the lower level messages that pass through each partner's protocol hierarchy.

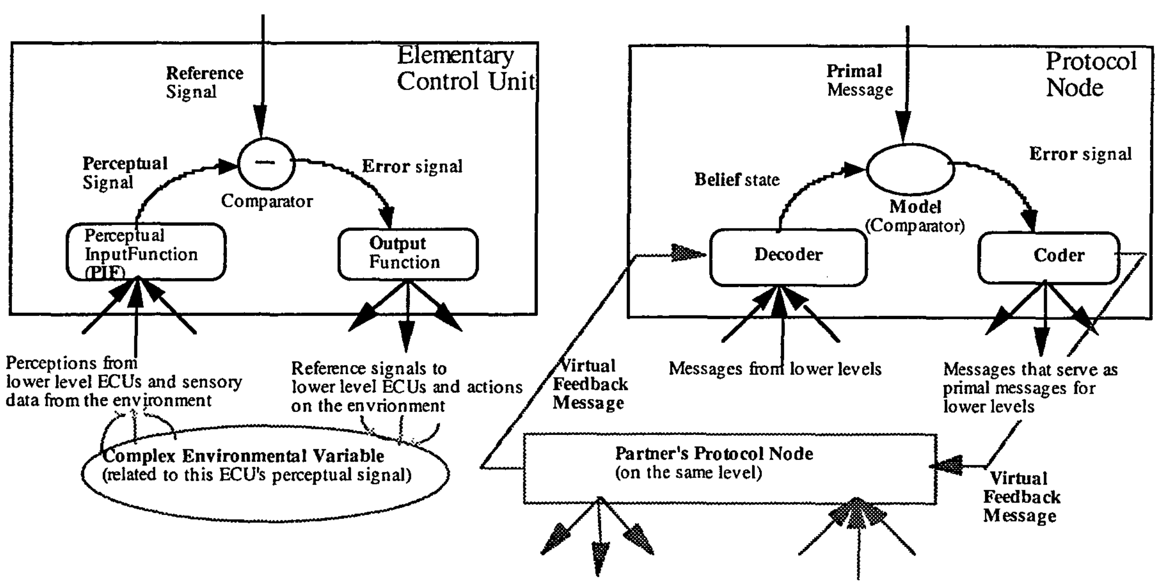


Figure 2. A comparison between PCT and LPT. The fundamental difference is that PCT is based on perception and LPT is based on the array of perceptions, or belief. LPT is also a tool designed specifically to interpret interactions between systems.

Since LPT is a subset of PCT, interaction and communication may be analysed using classical control methods. One of the difficulties of applying any mathematical analysis is the multi-dimensional nature of the belief signal that moves around the PN loop. PNs are connected hierarchically. Therefore the belief signal and the structure of the interaction are multi-dimensional

and presumed to be nonlinear. A mathematical analysis of the problem may be formidable, but not impossible for simple human systems.

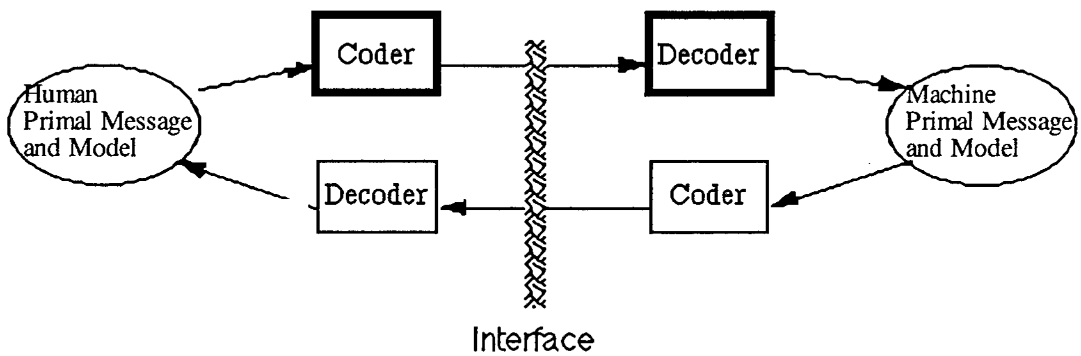


Figure 3. The connection between the Coders and Decoders of the Human-Machine interaction. Virtual messages are shown explicitly for a single level of abstraction.

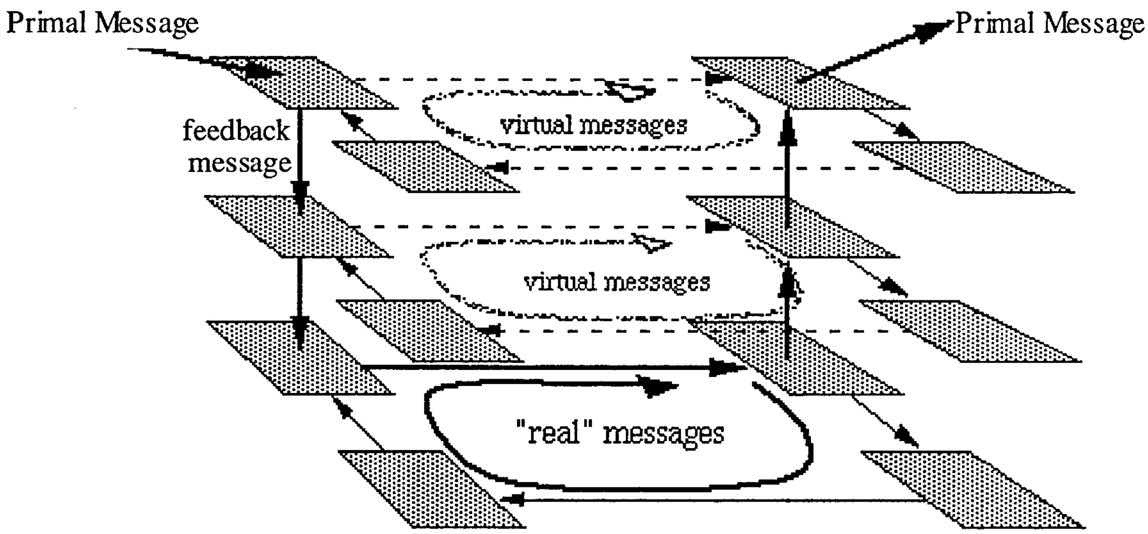


Figure 4. The relationship between the Primal messages, Virtual messages, Feedback messages, and Real messages.

Alternatively, a descriptive analysis is explored in this paper where the Primal message, Feedback messages, and the form of feedback are described with words for each PN. For example, the Primal message, *I want my hunger to be satisfied!* might be annotated within the Model oval of Figure 3. This statement represents the desired belief state. Determining the current belief state requires the culmination of lower level beliefs (perceptions) related to hunger, such as taste, smell, etc. At the lowest level of abstraction one can describe hunger in terms of chemical imbalances within the digestive system sending signals to the brain via neural impulses. At this level, mathematical equations might be used to describe the neuro-chemical reactions. However, one can envision the mathematical complexity in describing the lowest level perceptions, combining those perceptions into a single belief, and then deriving the operations that merge several beliefs to obtain the current belief state. Figure 4 illustrates how the hierarchical model might look.

Returning to the top level of this example, the different Feedback messages coming into the Decoder of Figure 3, such as *I have stomach pains!, It's 1200!, I'm salivating!*, etc. together establish the current belief state of hunger satisfaction. If the current belief state does not match the Primal message, then the Coder of Figure 3 forms appropriate messages to be transmitted to the communicating partner (in this example the partner could be a vending machine, another human, or the even the same person). The messages emanating from the Coder like *press candy button* or *start cooking* are Feedback messages for the partner's Decoder. Note that the Feedback messages in and out of the Decoder and Coder are themselves *Primary* messages for lower level protocol nodes. Again a hierarchy of beliefs and perceptions are generated, and these propagate downward to the physical level of abstraction where vibrations and pressure waves are generated at the mouth, moving air molecules. The sound waves reach the receptors within the inner ear where the sensory information is translated into perceptions of intonation, then words, sentences, ideas, and then beliefs at higher levels of abstraction.

The form of the Feedback messages for the Decoder and Coder depend on the belief states of both partners during the transmission of the Primal message. The General Protocol Grammar (GPG) is a set of 47 most probable forms of feedback. The GPG was defined by Taylor and Waugh (1991) to assist in recognising the forms of feedback necessary to convey the Primary message. Once the feedback required for successful transmission of the Primal message is determined, the Primary messages for lower level PNs are identified by the form of the feedback.

For example, if a person wanted their hunger to be satisfied, an overt form of feedback might be used to inform their partner of this desire. *Please get food* may take on a verbal form of feedback. The partners have, at least, a weak belief that they know what each other wants. On the other hand, a covert form of feedback might be employed when both partners have very strong beliefs that one is hungry and the other instinctively provides food without verbalising the request. All that might be required is the writing of a grocery list or perhaps no action (with respect to the Primal message) if, for example, it is after dinner and hunger has already been satisfied.

2.3 Pilot-CDU Interaction

The LPT framework can be applied to human-machine interaction, and in particular, pilot-CDU interaction. For example, the pilot may wish to establish a communication link between the aircraft and a ground station. The CDU may be considered as the pilot's partner who wants to believe that a radio link has been established. The LPT analysis should yield the requirements for feedback for the successful transmission of the Primal message. The requirements can then be compared to the current system from which the interface deficiencies may be determined.

For the analyst, it is sometimes difficult to imagine that the CDU comprehends messages at the higher levels of abstraction such as, *I would like to establish a radio link* or *I've chosen the appropriate radio.* It might be easier to imagine lower level messages such as *power on* or *radio 3.* Attempting to describe even lower levels, such as impact forces, stiction, photons and screen energy absorption rates, may not be necessary when proposed changes may be made only at the level of software implementation. The choice of where to begin and end the levels of abstraction depends on which aspects of the interface the analyst wants to explore. In this case, the highest level of abstraction is the pilot's desire to see a radio link set and the CDU's desire to satisfy the pilot. The lowest level of abstraction is defined as the messages that are designed and interpreted by the displays and controls of the CDU.

It is hypothesised that a Layered Protocol analysis will lead to the requirements for an interface that addresses the interaction deficiencies. That is, if a particular key does not convey the required Virtual message, or if a particular display interferes with other necessary information then the designer must redesign the interface to ensure that the proper Virtual messages are available for each level of abstraction. It is important to note that the LPT analysis may yield required Virtual messages but DOES NOT make any inferences on how to implement them into a coherent interface.

3. THE LAYERED PROTOCOL TOOL

The Layered Protocol Tool (LPTool) was developed under contract for DCIEM (contract no. W7711-4-7226/01-XSE). It is a software program that allows the user to generate, name, and annotate icons that represent protocol nodes and their different views. The protocol node icons can be connected to each other and a software routine checks for proper connectivity between nodes. The resultant model is a hierarchical structure of levels of protocols that constitute the Primal message. There is no capability within the software, currently, to simulate the passage of Virtual messages to and from the communicating partner. For a complete description of LPTool and its views see Farrell, Hollands, and Taylor (1997).

The analyst may choose to begin generating a model from either the pilot's or CDU's point of view. It seems more natural for the analyst to put himself in the place of a pilot and begin the analysis. However the number of probable protocols are significantly less when describing the interaction from the CDU's point of view. For this report, the analyst takes on the role of the CDU.

3.1 Network View

A Protocol Node (PN) has a polygon shape with three letters and four quadrants as shown in Figure 5. Two icons, representing transmitting and receiving PNs, and an arrow icon appear on startup of the program. A transmitting node describes the Primary message that is sent from the originator (i.e., CDU) to the recipient (i.e., pilot). A receiving node describes the Primary message that is sent from the recipient to the originator. Feedback messages going into a PN's Decoder are the Primary messages from receiving PNs at the next level down. Feedback messages leaving a PN's Coder become the Primary messages for transmitting PNs at the next level down.

A protocol node is generated by highlighting either the transmitting or receiving PN icon and then selecting its position on the Network View window. The PN's Model is opened by double clicking on the letter *M*. A window appears with the buttons Insert, Delete, Edit, Annotate and OK. This allows the analyst to define and describe, in words, the Primary message(s) associated with the PN. Similar windows appear when the PN's Coder (C) or Decoder (D) is double clicked, and the analyst may annotate the expected Feedback messages to be sent and received.

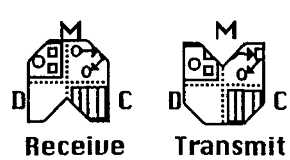


Figure 5. Transmit PN depicts Primary messages emanating from within the originator. Receive PN depicts Primary messages emanating from within the recipient.

Connections between protocol nodes are made by depressing the command key, selecting one of the protocol's letters, and then dragging and releasing at the desired letter of the second protocol (e.g., connecting between D and M, or C and M). An algorithm checks the connectivity between connecting PNs.

Three of the four PN quadrants activate views onto the Nine Element View, the General Protocol Grammar (GPG), and the Job Processing Chart. The Job Processing Chart and the fourth quadrant are not currently functional but have been identified for simulation purposes. The Nine Element View and the GPG are discussed in limited detail below. For more complete descriptions refer to Farrell et. al. (1997) and Taylor, Farrell, Hollands, and Semprie (1997).

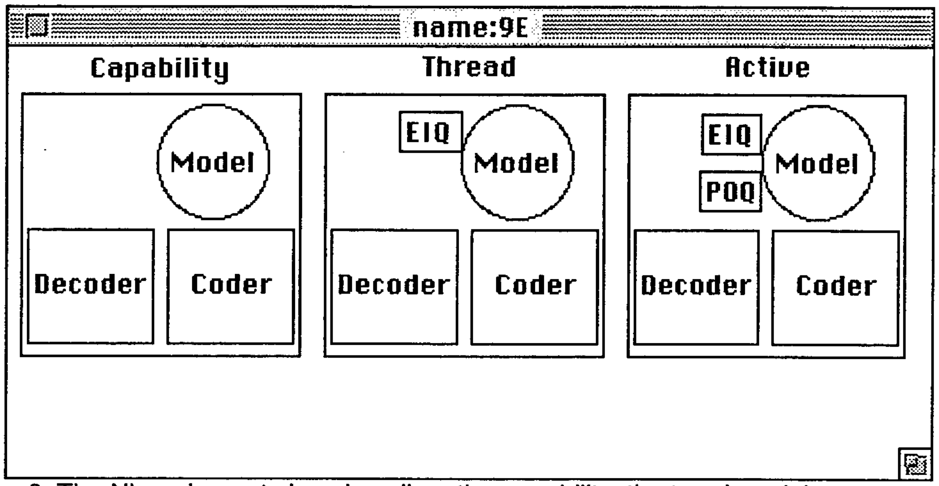


Figure 6. The Nine-element view describes the capability, the trend, and the current state of a protocol node.

3.2 Nine Element View

The Nine Element View is opened by double clicking the upper-left corner of the PN. The Nine Element View in Figure 6 depicts three time slices of the protocol model; Capability, Thread and Active. The Capability slice describes the quasi-permanent capabilities of the Model, Coder, and Decoder. The Thread slice represents a recent history of states of the PN, and is used to predict how the interaction might proceed in the very near future. The Active slice holds the current state of the specific dialogue at a specific moment. The Expected Input Queue (EIQ) and the Predicted Output Queue (POQ) provide links to other windows in the Network. The Nine-Element View will become critical in a future version of the LPTool that incorporates dynamic simulation of the passing of messages and levels of beliefs.

3.3 General Protocol Grammar

Taylor et al. (1997) provides a comprehensive description of the General Protocol Grammar (GPG). The following section gives only an overview of the grammar. The GPG represents evolving belief states of both partners

about a Primary message, from the originator's perspective. The goal of the originator is to believe that the recipient has adequately interpreted the Primary message. The goal of the recipient (in cooperative communication) is to adequately interpret the Primary message. Ideally, once the goals are achieved, it is not worth continuing. From these statements of fact, three propositions were defined about the belief states of both partners during the conversation:

P1: The recipient has made or is in the process of making an interpretation of the primal message.

P2: The quality of the communication mechanism is sufficient for an adequate interpretation of a message.

P3: It is not worth continuing to improve the recipient's interpretation of this message.

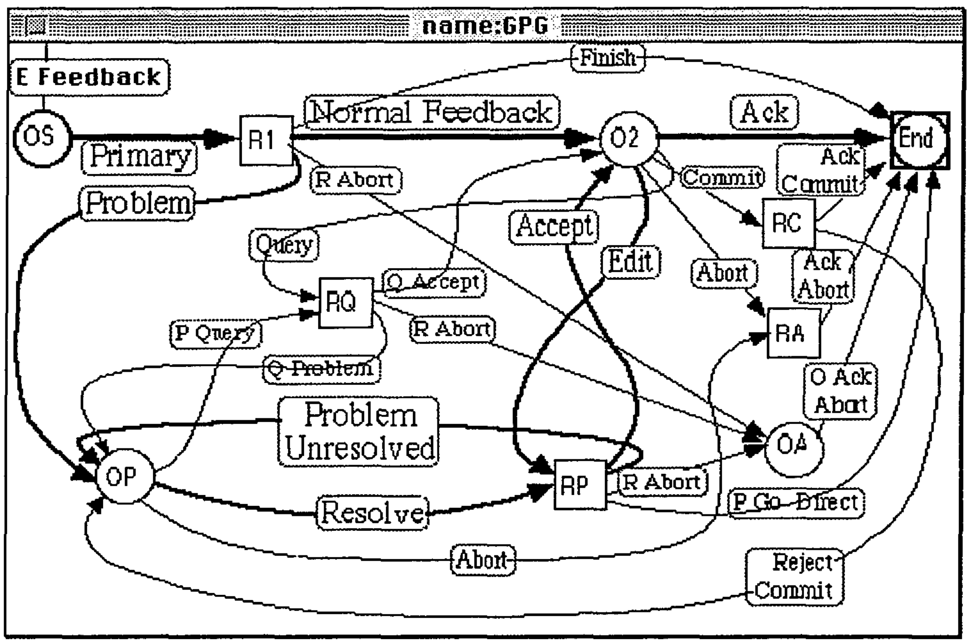


Figure 7. The GPG describes 47 different paths of possible communication that occur in most interactions. Each path can be opened to a view of possible forms of feedback that the path may take on. The analyst may enable or disable the path and its instantiation[1].

The nodes in Figure 7 represent particular combinations of beliefs about the three propositions. The arcs represent Feedback messages required to move one's partner from a current belief state closer to the desired believe state. Taylor et. al. (1997) describes all the ideal belief states of the nodes and arcs in the GPG with respect to the partners' beliefs about the three propositions.

The GPG view looks like a state transition diagram. However, it is far from that, since the partners' beliefs may change smoothly or abruptly. The GPG provides snapshots of the more probable beliefs states that may occur during

[1] *Instantiation* is defined here as the form of a message. *To instantiate* means to assign some form to a message.

the passage of the Primal message. For example, by definition of P1, the originator's belief state may have membership at OS, Primary Arc, and R1 all at the same time. In fact, every node and arc have some level of belief associated with the three propositions at every instant in time, albeit some would be more prominent than others at different times in the conversation.

In addition to the originator's belief, the originator has some belief about what the recipient believes about the three propositions. This is a second recursion of belief and it determines the form of feedback that the originator will give to the recipient. For example, if the originator believes that the recipient believes *not P1* then the originator would provide some overt feedback to the recipient about the Primary message. If the originator believes that the recipient also believes *P1* then no feedback is necessary. A third recursion of belief is required when it is the recipient's turn to send a message. That is, with respect to the originator's GPG, the recipient needs to have some belief about what the originator believes about the propositions in order to determine the required form of feedback.

Three forms of feedback have been identified for transmitting messages: Null, Neutral and Inform. Null feedback is invoked when both partners believe that they are in the desired belief state, and so no overt form of feedback is necessary. Neutral feedback is defined as a Feedback message that does not contain any content of the Primary message. Neutral feedback may be something like, *uh?*, or *OK*, or a facial expression, etc.. Inform feedback is a message that includes part or all of the content of the Primary message. Inform feedback is most often used when the error between the actual and desired belief states is great.

Inform feedback is expressed in several forms in the LPTool including Verify, Correction, Propose, and Enquire. Verify feedback restates all the content of the Primary message. For example, if the originator's message was, *I want some ice cream*, the recipient might reply, *You want some ice cream.* With this feedback, the originator may come to believe that the recipient has a strong belief in *P1* and strong belief *P2*. If the period is replaced with a question mark then the implication is that *P2* is believed weakly and the Enquire feedback may be used. Verify feedback is defined within the Normal Feedback and Accept arcs, while Enquire is found in the Query arcs.

Correction feedback occurs when there is a strong belief that a contextual error has been made. For example, the originator says *I want some ice cream*, and the recipient might say, *I see, you want some sour cream.* Note that the message protocol remains constant while the message content has changed. Correction feedback is found within the Normal Feedback and the Accept arcs.

Propose feedback is used in situations where the current Primary message is inadequate in the context of the overall conversation. An attempt is made to restructure the Primary message. For example, the originator says, *I want some ice cream*, and the recipient might say, *You mean you want to go for a*

jog before your snack. The recipient tries to alter the originator's Primary message by presenting other options. Propose feedback is found within the Problem and Problem Unresolved arcs.

Appendix A provides a complete description of the nodes, arcs, and feedback forms in terms of the three propositions. It is necessary to refer to this reference appendix for understanding the following appendices where the GPG is specified and simplified for the pilot-CDU interaction.

An analyst must step through the GPG and determine the most likely arcs and forms of feedback during the transmission of the Primary message. Once the forms of feedback are identified, the analyst must determine the protocol nodes that would support that form of feedback at the next level down. Inform and Neutral feedback require supporting protocol nodes, while Null feedback does not. As the GPG analysis evolves, it is soon evident that many of the lower level protocols are identical and can be multiplexed (Farrell et. al. 1997).

Analysing the GPG is time consuming, but in practice only a small portion of arcs need to be instantiated. For instance, at the level of displaying the results of a keystroke, the CDU may provide a Primary message by displaying the letter *a*, for example, on the screen. The pilot is not given an opportunity to abort, provide normal feedback, or determine whether there is a problem with the displayed letter. Therefore, the Finish arc is used and all other arcs are disabled for this very low level protocol.

Despite the effort put into the LPTool development, the tool had not been applied to any real system. Requests for advice on difficulties experienced with the CH-146 Griffon helicopter CDU provided the opportunity to use the LPTool on a practical problem.

4. ANALYSIS: THE CH-146 CDU

The CDU used for the Canadian Forces CH-146 Griffon helicopter was developed by Canadian Marconi Company (CMC). A schematic of the CDU is shown in Figure 8.

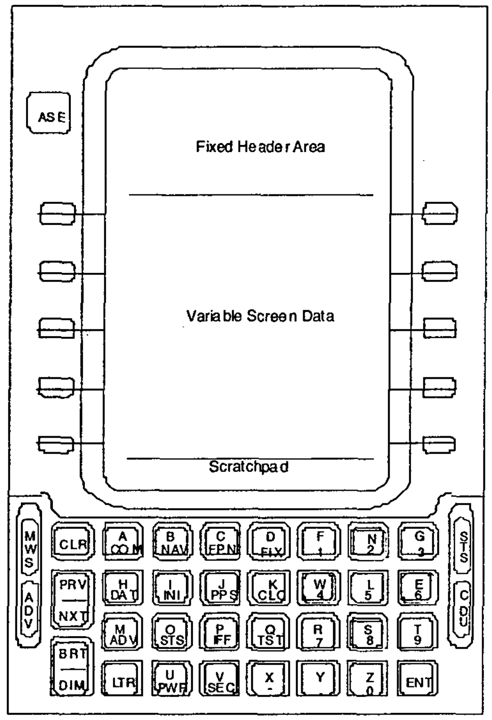


Figure 8. The CDU displays mission and system data and permits the operator entry and modification of mission data. The CDU also provides information exchange between the flight crew and the CH-146 avionics sub systems. The CDU provides a dot matrix, thin-film electroluminescent (TFEL) display, mounted with a full keyboard composed of 29 alphanumeric keys, two rocker keys, ten display adjacent software-programmable keys (soft keys), and four enunciator keys (reprinted with permission from Canadian Marconi Company).

A list of problems were identified in a letter to DCIEM from LATEF OLA section. The problems included:

1) the operator's inability to determine if the radio coordinates that are entered into the scratch pad have in fact been acknowledged by the appropriate radios,

2) the system engagement of options as they are cycled through which sometimes leads into lengthy initialisation processes that can adversely affect flight,

3) the mechanisms involved in calculations of ground speeds which are based on information that becomes obsolete when in flight, and

4) several complaints regarding the physical design of the interface.

Problems 1, 2, and 4 address deficiencies with respect to the pilot-CDU interaction. Layered Protocol Theory was identified as a tool that could analyse the interaction and possibly determine the causes of, what is essentially, the break down in communication. The intent was to begin with a simple task, such as establishing a radio link, and model the interaction using LPT methods.

4.1 A Brief History of the CDU Analysis

No guidelines for this analysis were available for this first attempt at analysing a real device using the LPTool. A detailed written log was kept during the analysis. These first steps are summarised in this section. The words in italics indicate the name of the LPTool document. Appendix B contains a listing of Network Views of all the models developed throughout the analysis.

The analysis began at a very high level of abstraction where the pilot wanted to believe that the pilot was in flight. The LPTool files called *marc* and *marc2* contained the first attempts at building an interaction model. However, the model turned out to describe the aircraft components and not the interaction between the pilot and the aircraft.

The human-machine interaction was not immediately intuitive. The Layered Protocol theory was re-visited, using a human-human interaction example, shown in *client_hotel,* in order to gain some insight in using the LPTool. The resultant model described the client's and hotel clerk's models on the same Network View. It was quickly learned that the transmitting and receiving nodes had specific meanings with respect to which partner the analyst wanted to model. From these observations, it was clear that Feedback messages coming from the Coder were to be connected to the Model of a transmitting PN, and Feedback messages going into the Decoder were to be connected to the Model of a receiving PN.

The flight protocol was then analysed with a clearer understanding of the relationships between transmitting and receiving protocol nodes. In the *flight* series of models, it was assumed that the pilot was both originator and recipient of messages at the top levels of abstraction. In *flight2a,* the GPG was completed, yielding lower level, supporting protocol nodes. The communication and navigation protocol nodes were expanded in *flight2b* to determine if the GPG analysis would yield the CDU protocol.

The *flight3* series of models showed a hierarchy of PNs from the top level of wanting to believe that the pilot was flying to the level of wanting to perceive navigation and communication instruments that would aid in the flight belief. A separate model was generated called *commnav* that explored only the communication and navigation protocol nodes being supported by a CDU protocol node. The *commnav* model was added to the *flight3* model producing *flight3c.*

The analysis then considered the pilot as the originator and the CDU as the recipient of the Primal message. A screen definition document (Bell, 1995) was used to interpret the controls and displays as messages, and generate the appropriate protocol nodes. However, the resultant model seemed to describe the relationship between components of the CDU screens rather than the interaction between the pilot and the CDU. This situation transpired because the PNs' GPGs were not annotated one level at a time and at every level.

Listed below are the lessons learned for the construction of a Layered Protocol model:

- Start and end at levels where a designer may affect changes.
- Define the partner for which the model is to be developed.
- Ensure that Coders are connected to transmitting nodes.
- Ensure that Decoders are connected to receiving nodes.
- Annotate completely the GPG before generating and annotating supporting nodes.

4.2 CDU-Pilot Interaction Model

For the Layered Protocol interaction model, the pilot and CDU are considered to be communicative partners. The model yields a description of probable belief states and Feedback messages as the originator is relaying a Primal message to the recipient. A LPTool model is built from the perspective of one partner. In this case, the model was developed from the CDU's perspective since there is a finite feasible set of mechanisms with which the interaction takes place through the CDU interface. Therefore, the CDU is the originator of the Primal message and the pilot is the recipient. Giving the CDU a personae may help the analyst to think in more natural terms of human-human communication. Note that a pilot's perspective model would mirror a CDU's perspective model (see Figure 4).

4.3 Top Level Protocol Node and the Primal Message

The CDU-Pilot interaction model begins with a statement of the Primal message. The CDU might say, *I want to believe that a radio link is set*, or *I want to believe that a waypoint is set*. Note that, although the specific context is one of either communication or navigation, only a single protocol is necessary to describe the message structure (i.e., *I want to believe **something** is set*).

A transmitting PN icon is selected from the LPTool palette, and placed on the Network View window. It is labelled *CDU*. Double clicking the *M* icon opens the Capability Model view where the Primal message is titled *communication* and annotated within another window as shown in Figure 9. A similar message is annotated for *navigation* where the CDU wants to believe that a waypoint is established. The Primal messages *communication* and *navigation* emanate from the Coder of higher level protocol nodes and are multiplexed into this single protocol node.

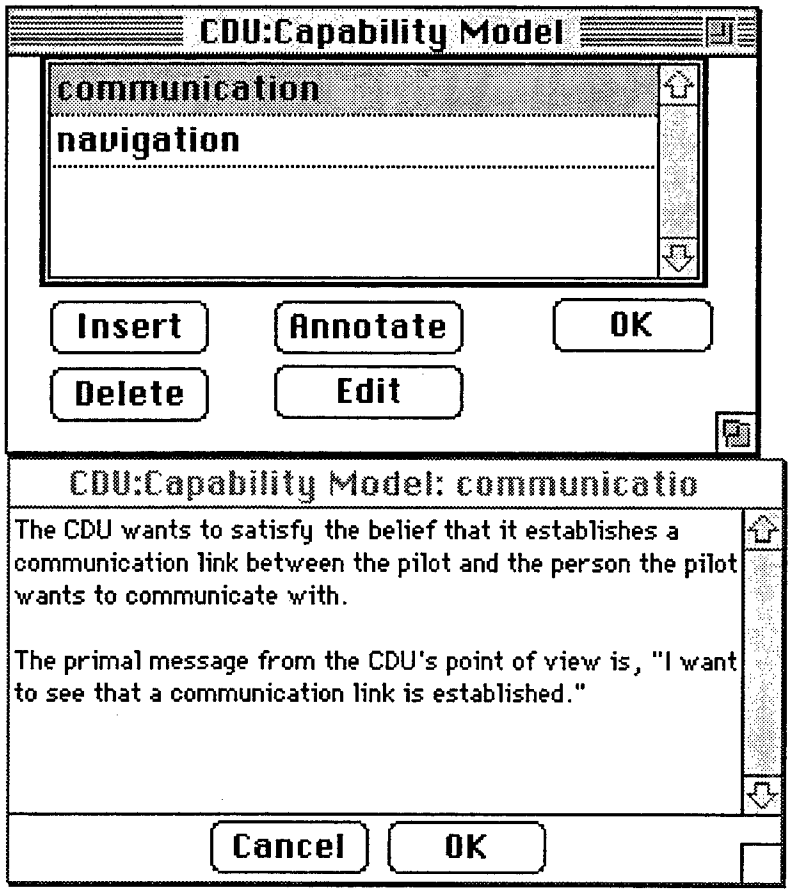


Figure 9. The CDU:Capability Model view is generated by clicking on the letter *M* of the PN. Insert, Delete, Annotate, Edit, and OK are functional buttons for the generation of the primal message. Clicking on the *communication* Primal message opens the window, CDU:Capability Model:communication. Within this window the analyst may describe the details of the Primal message.

4.4 GPG Annotation

The General Protocol Grammar for the CDU protocol node was opened and examined next. The analyst was, momentarily, not interested in the content of the Primal message, but whether the Primal message had been interpreted and understood. The GPG assists the analyst in keeping track of the Feedback messages and the belief states between the CDU and the pilot for successful transmission of the Primal message. For all arcs in the GPG the analyst must ask the same question: that is, which are the most likely forms of feedback during the transmission of the Primal message?

The first arc in the grammar is the *E-feedback* arc which provides information about the current state of the pilot/aircraft system. It includes information about the physical state of the aircraft and environmental systems as well as the pilot's current belief state. Due to an oversight in designing the program, *E-feedback* is not shown in Figure 10. However, *E-feedback* was considered in this model. Double clicking the *E-feedback* arc yielded a window on possible forms of Feedback messages that contained information about the pilot/

aircraft states. Inform feedback was chosen to be the most probable form at any given time throughout the interaction.

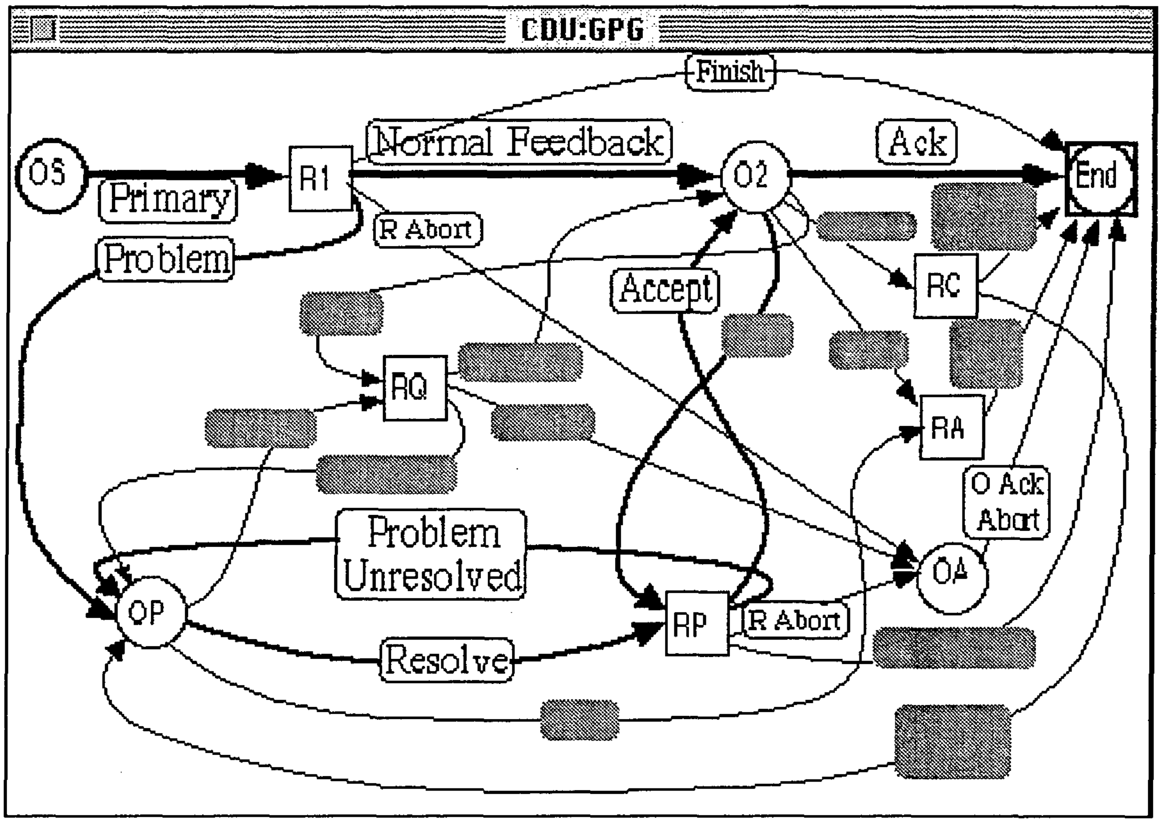


Figure 10. The GPG view is generated by clicking the upper right quadrant of the PN. The E-feedback arc is missing for this transmitting node. The arcs are enabled or disabled by selecting the arc and then toggling a menu item.

The CDU's *Primary* arc would follow the *E-feedback* arc in Figure 10. The *Primary* arc represents the first passage of the Primal message at this first level (or represents a Primary message if the PN is a supporting node) from the Originator at OS to the Recipient at R1. The *Primary* arc, however, does not provide the content of the Primal message but simply the possible forms of Feedback messages.

As before, double clicking the *Primary* arc opens a view onto the possible instantiations for this arc as shown in Figure 11. In this case, the CDU is transmitting a message to the pilot and both the Inform and Null instantiations were chosen. For Inform, the CDU provides an overt Virtual message that it wants to establish a communication link by displaying the related components such as the radio names, modes, and frequencies. At other times, both communicating partners are aware of the purpose of the CDU, or a link is already established, and thus the Null form of feedback was enabled. Note that an overt form of feedback automatically requires Primary messages within supporting protocols at the next level down.

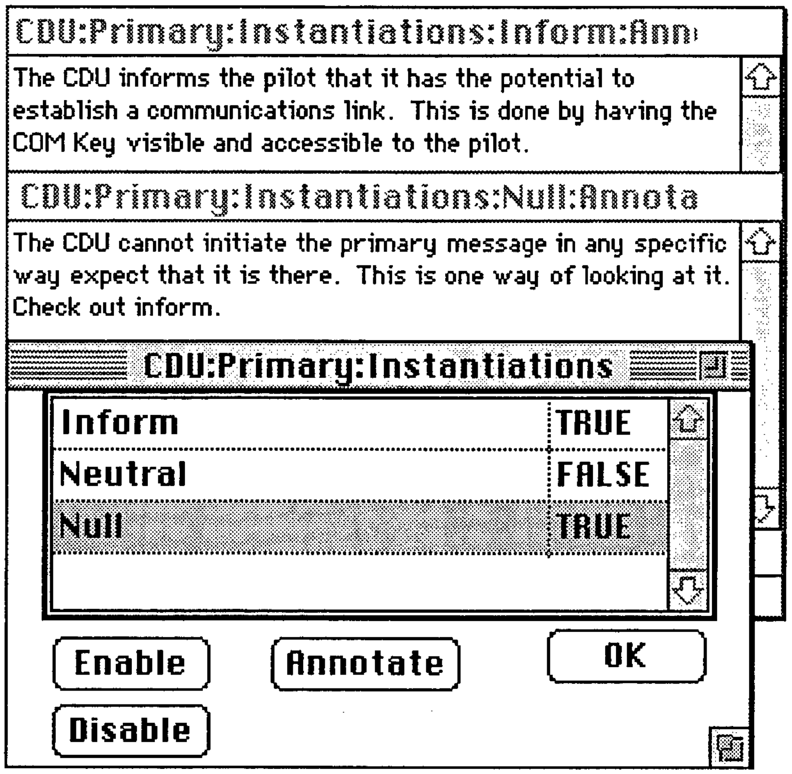


Figure 11. The CDU:Primary: Instantiations view is generated by clicking on the Primary arc of the GPG. The analyst may enable or disable and annotate the different instantiations. In this case, both the Inform and the Null instantiations are enabled and their annotations are shown above.

Ideally at R1, the pilot has made an interpretation of the Primal message (i.e., P1), and initiates a Feedback message to the CDU based on current beliefs about P2 and P3. Four arcs emanate from R1 namely, *Finish, Normal Feedback, R Abort*, and *Problem*. All arcs were enabled as it was thought that the pilot could potentially be in any of the belief states represented by the arcs. Table 1 shows the relationship between the arcs and the ideal belief states.

Table 1.

Arc	Proposition	In words
Finish	P1 & P2 & P3	The pilot believes that the current and desired radio links are the same and ends the communication.
Normal Feedback	P1 & P2	The pilot confirms that that the current link is the desired one.
R Abort	P1 & P3	The pilot does not want to continue with this particular transmission for reasons not immediately identified.
Problem	P1 & not P2	The current link is not the desired one and the pilot transmits feedback to sort out the problem.

The *Finish* arc represents the pilot's desire to not continue the message because it has been adequately interpreted. Both the Null and Neutral forms of feedback were enabled for the *Finish* arc since the pilot could either recognise that the link was established by talking through the current radio (Neutral), or choose to ignore interacting with the CDU and radio systems (Null).

Null, Neutral, and Verify forms of feedback were enabled for the *Normal Feedback* arc. That is, the pilot either a) elected to not respond since both partners have a strong belief about P1 & P2 (Null), b) recognised that the Primal message is in process by paying attention to the CDU (Neutral), or c) verifying the content and interpretation of the Primal message by starting up, initialising, setting the radios, frequency, mode, etc. (Verify). Depending on the strength of belief about the *Normal Feedback* proposition (i.e., P2) at a particular moment in time, any one of the three instantiations are plausible. At O2, the CDU has received a response from the pilot and acknowledges that response by displaying the appropriate changes on the screen.

The *Acknowledge* arc reflects that the CDU's believes P1 & P2 & P3, and the CDU believes that the pilot believes P1 & P2 & P3. Therefore, the CDU ends the message transmission. It may seem that the CDU controls the interaction. Conversely, the CDU's belief state is in flux and indeed fuzzy, and alternates between the OS and END nodes as it is simply displaying information.

Each arc was enabled with specific forms of feedback. For instance, the Neutral form of feedback was enabled within the *R Abort* arc (e.g., the pilot turns off the CDU). Note that the supporting PN can be multiplexed onto a startup protocol node identified in the *Normal Feedback* and *Problem* arcs. At OA, the CDU may provide a Neutral form for acknowledging the abort (*O Ack Abort*) by changing its state (e.g., power light goes off). Again, a protocol node must be generated to support this message.

The *Problem* arc is necessary when the pilot has interpreted the primal message to be different from what is expected; in this case, *to see a communications link established*. For instance, the pilot may have just finished setting a waypoint. Now the CDU is displaying waypoint coordinates. Therefore, the pilot must inform the CDU that the Primal message was not adequately interpreted.

The Propose instantiation of the *Problem* arc was enabled so that the pilot may influence the content of the Primal message and attempt to convince the CDU to change it's belief about the Primal message. The pilot does so by starting up, initialising, setting the radios, frequency, mode, etc. Note that the supporting protocols are identical to those found in *Normal Feedback*.

At OP, the CDU may *Resolve* the problem and display a screen that corresponds to the Primal message. The supporting protocol nodes for *Resolve* are identical to the *Primary* arc's supporting nodes. At RP, the pilot may a) *Accept* the message similar to the *Normal Feedback* arc, b) transverse the *Problem Unresolved* arc which is similar to the Problem arc, or c) take the *R Abort*, all depending on the current level of belief of the Primal message with respect to the three propositions. It is quickly evident that many arcs within the GPG are identical, but are differentiated depending on the evolution of the belief state that each partner has of the Primal message. All annotations for the complete model are listed in Appendix C.

4.5 Supporting Protocol Nodes in the Network View

Supporting protocol nodes were derived for each arc within the GPG at the CDU level. That is, if the form of feedback was overt, then a PN was required to support that Feedback message. For example, if a frequency setting was required to complete the Primal message of establishing a radio link, then the desired belief state about the frequency setting became a Primary message for a lower level protocol node. The analysis showed that the frequency setting supported many arcs in the CDU protocol. Only one frequency protocol was required for each of the Feedback messages that required it.

Furthermore, it became evident that many of the arcs had complimentary arcs within the same GPG. For example, for the CDU protocol, the *Primary* arc's message was that the CDU wants to establish a radio link. Its complementary arc was the *Normal Feedback* arc where the pilot wants to see a specific radio link established. Both arcs required a lower level radio setting protocol; one for the CDU to display the radio settings (FUNCTION transmitting node) and one for the pilot to enter in the radio settings (FUNCTION receiving node). Such complementary nodes were common within the Network View.

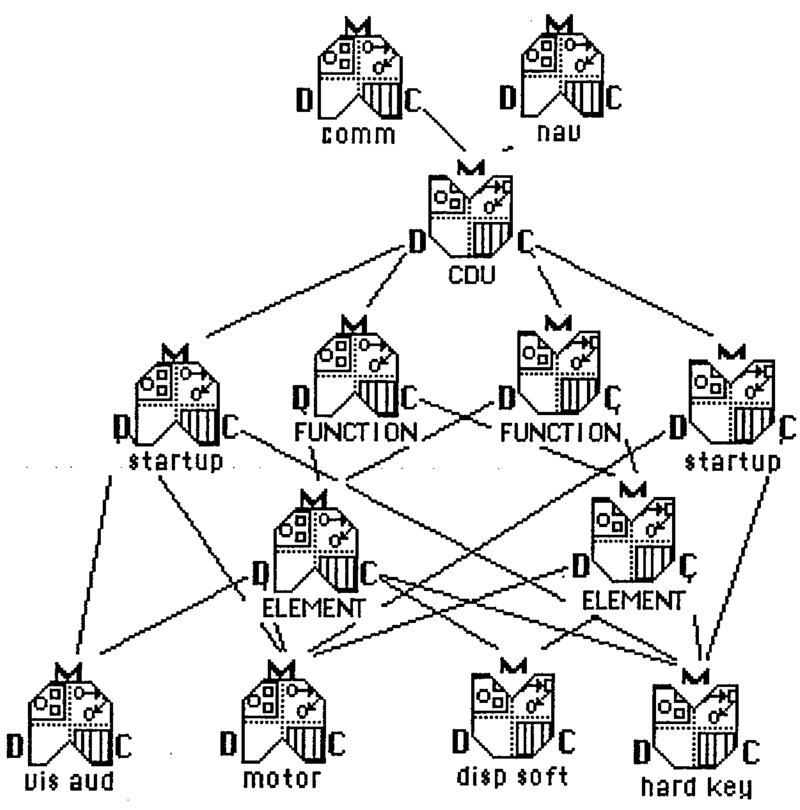


Figure 12. Network View of the pilot-CDU Interaction Model. Subsequent levels of protocol nodes are derived from a detailed analysis of the GPG.

Once a unique set of Primary messages that support the arcs within the GPG are established, the messages that originate with the CDU are listed in the Coder, and those that are received by the CDU are listed in the Decoder. New

protocol nodes are generated and links are made from the Coder of the *CDU* protocol node to the Model of the supporting nodes, *FUNCTION* and *startup*, as shown in Figure 12. These two protocols had significantly different GPG structures and warrant having their own protocol node.

An identical analysis is required for each supporting node, starting with a definition of its Primary message, a GPG analysis, and supporting PN identification. For example, the transmitting *FUNCTION* node analysis was completed and yielded Feedback messages that identified the elements needed to establish a link (i.e., radio type, frequency, mode, and security). These messages have similar protocols and were multiplexed onto a single PN called *ELEMENT*. Each *ELEMENT* message must be complete in full before the communication function is complete and the link is established.

The GPG of the *ELEMENT* protocol was analysed and yielded supporting protocols that required the pilot to interact with the physical interface of the CDU and visa versa. At this low level of abstraction, the pilot receives messages with their eyes and/or ears, and transmits messages with their fingers and/or voice depending on the details of the interface design. If the model were to continue, then the analyst would look at impact forces on the hardkeys and light levels from the display that are required to complete *hard key* and *visual/audio* messages, etc..

4.6 Pilot-CDU Interaction Deficiencies

An interaction is deficient when there is an inability to effectively determine and modify a belief state of either partner. That is, an arc within a GPG might be missing, incorrectly enabled or disabled, or redundant leading to slowly stabilising or unstable belief states. Ultimately, this situation will lead to a break down in communication. The deficiencies in the analysis were found by listing the form of feedback for an arc within a PN's GPG and the way it was implemented within the actual CDU interface. A listing of the instantiations are found in Appendix D. Table 2 is a sample of Appendix D. Alongside the current instantiations is a column for proposed changes to the interface that would provide the appropriate feedback.

An example of a disabled arc, that could very well be enabled to ensure successful message transmission, is the Abort arc within the receiving FUNCTION protocol. Many of the functions' options are toggled in the current CDU design. As an option appears on the screen, it is activated! The pilot can not simply view the options without activating them. In a proposed CDU design, the active option and the selected option are two different entities and may be displayed at all times. The pilot may select an option but to activate that option they must press the enter key. The act of selecting another option, without activating it, is equivalent to aborting the previously selected option. This observation is related to the first and second deficiencies mentioned in the LATEF OLA reference.

Table 2.

Receiving FUNCTION	"The Pilot wants to see all completed elements"	
form of feedback	*Current interface implementation*	*Proposed interface implementation*
E-FEEDBACK		
Inform	*deficient*	All editable elements are to be displayed simultaneously
PRIMARY		
Null		
Inform	appropriate fields and menus are available for the pilot to edit using the soft keys	appropriate fields and one nested menu are available for the pilot to edit using the rocker key
NORMAL FEEDBACK		
Verify	CDU highlights element being edited by changing screens and changing field background colour.	see E-FEEDBACK.
EDIT		
Inform	see PRIMARY.	see PRIMARY. The pop up menu assists the editing by listing all states including the current one.
ACCEPT		
Verify	see NORMAL FEEDBACK	see NORMAL FEEDBACK
ACKNOWLEDGE		
Neutral	*ambiguous*	clicking proposed **ent** saves element
ABORT		
Neutral	*deficient arc*	no **ent** defaults to current state
ACK ABORT		
Neutral	*deficient arc*	current state is displayed

The simultaneous display of the active and selected options provide a means of comparison concerning what CDU and pilot believes about a current option. One can expand this idea to having an expected option field where the CDU may provide a list of most probable options for that radio link. This might have proved beneficial in the Cali accident if the most likely waypoints appeared beside the current ones being selected. If an NDB was selected by the pilot that was not anticipated by the CDU, it might display both the active, selected and most likely beacon, thus forcing the pilot to resolve any potential discrepancies.

The third deficiency observation from LATEF is with respect to the communication between the CDU and the aircraft dynamics. This introduces a new area of study which is analysing machine-machine interactions using Layered Protocol Theory.

The designed interface described in the next section addresses the fourth deficiency observation mentioned in the LATEF OLA reference. The layout captures all the required Feedback messages between the CDU and the pilot at all levels of the hierarchy without clutter, and with only one level of nested menus. It is also proposed that the INI, STS, and POW functions operate in the background, either on startup or when a radio link is selected.

Although this paper deals with the Layered Protocol Analysis of the Control Display Unit, it is important to document the shortcomings of the LPTool program itself as well as possible solutions for future versions. A list of limitations was compiled and is presented in Appendix E.

One last observation is that the Network View also suggests a disconnect between the number of levels of abstraction and the number of displays and controls that are currently part of the CDU. The CDU incorporates 40 screens, 10 soft keys, and 31 hardkeys related to establishing a communication link. However, the Network View shows only ten unique protocols. A good design of the CDU interface might yield a closer mapping of the protocols to the number of displays and controls needed to transmit the information.

4.7 From LPT Model to Interface Design

The Layered Protocol Model yields the required Virtual messages at each level of abstraction that ensures the efficient passage of the Primal message. A designer of the CDU interface might say that the Virtual messages are guaranteed to be interpreted adequately if the appropriate displays and controls are incorporated in the interface design.

Appendix D describes the implementation of the current displays and controls that make the transmission of a Virtual message possible. The third column in each table also lists the proposed changes to the interface in order to adequately transmit the message. Each message was studied in turn and the appropriate controls and displays were incorporated in the interface and the philosophy of its use.

However, once the controls' and displays' information and action requirements are determined, the layout of the interface is still somewhat of an art form. The designer must use visual interface design techniques (Mullet and Sano, 1997) to perform the trade offs between space constraints and the amount of information being displayed at any given time. It is critical, however, that the Virtual messages at each level of abstraction are clearly articulated within the interface.

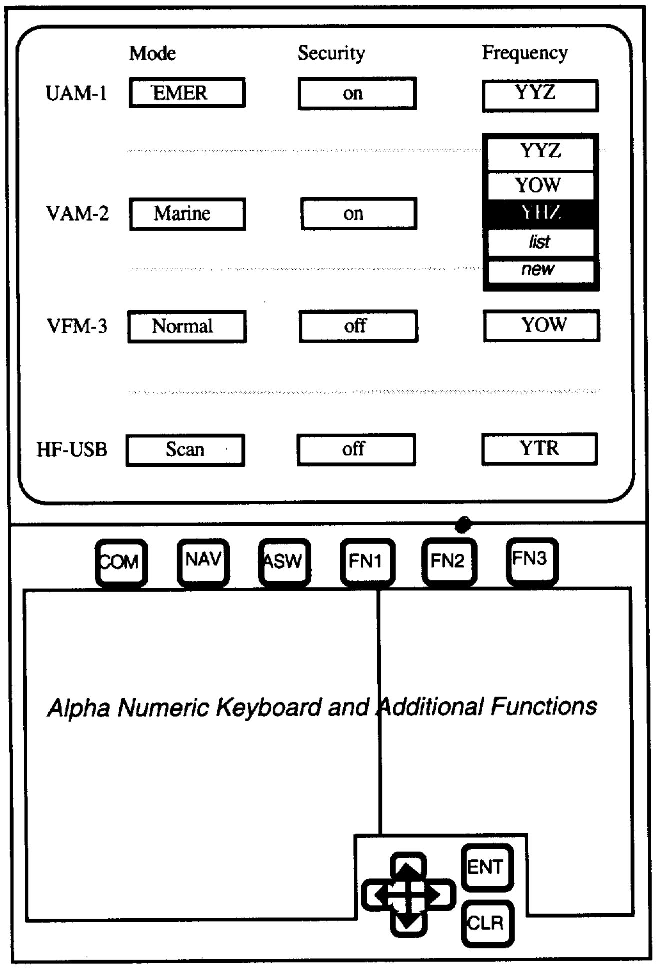


Figure 13. One possible layout for the CDU interface that incorporates on a single screen and one nested menu structure all the necessary information for establishing a communications link.

The highest level of abstraction (i.e., the desire to assist in communication and navigation) is satisfied by the look and feel of the interface. It indicates that the pilot is dealing with a CDU rather than a calculator or telephone. At the next level of abstraction (i.e., establishing a communications link), pressing the COM hard key displays a screen similar to the one shown in Figure 13. This screen provides information about the four communication links. The next level of abstraction (i.e., setting, individually, the radio, mode, security, and frequency) is clearly delineated by each column of the matrix. The rocker key and enter key provides a means to toggle amongst the matrix positions. The final level of abstraction (i.e., specifying the elements) can be done by selecting the desired state from a pull down menu at each of the matrix positions. To illustrate that multiple design solutions may exist, two other designs have been made that incorporate some of the constraints of the real CDU system. However, in all cases, the protocols at each level of abstraction are represented in each design.

5. SUMMARY AND CONCLUSIONS

A layered Protocol analysis of a Control Display Unit was performed using a new software program called LPTool. The technique was applied to the interaction between the pilot and the CDU as partners in establishing a radio communication link. The CDU-Pilot interaction was modelled in detail, starting with a definition of the Primal message, determining the forms of feedback within the General Protocol Grammar, and identifying supporting protocol nodes for the next level down. The analysis yielded deficiencies which paralleled those identified by LATEF. The interaction deficiencies were listed, and a proposed layout was presented that addressed the deficiencies.

The proposed interface layout provided feedback on the current, desired and expected radio communications links. This information is required to determine both the CDU's and pilot's belief states and attempt to match them to their own references. The LPT model does not provide a method for designing an interface but yielded the necessary messages that are critical for successful transmission and understanding of the primal message.

The LPTool program assisted in identifying the required Feedback messages during the interaction. An analyst or designer may use the tool to organise plausible feedback messages at many levels of abstraction as one envisions the interaction between the user and the machine. However, LPTool itself is awkward to use and not optimised. A future version of the tool might address these problems as well as provide a tutorial for using Layered Protocol Theory for interface design.

This Page Intentionaly
Left Blank

6. REFERENCES

Bell Helicopter Textron, (1995). Canadian Forces Utility Tactical Transport Helicopter Program: Software Requirements Document. (Contract No. 826000, Document Number No. 147-a60331-005, CDRL #SE-323). Kanata, Ontario, Canada: Canadian Marconi Company.

Farrell, P.S.E., Hollands, J.G., & Taylor M.M., (1997). Perceptual Control and Layered Protocols in Interface Design. I. Fundamental Concepts. (in progress).

Hendy, K. C. (1994). An information processing model for workload and performance prediction. In, K. C. Hendy (Eds.), *Proceedings of the 'Workshop on Task Network Simulation for Human-Machine System Design'*. Washington, DC, USA: The Technical Cooperation Program, Subgroup U, Technical Panel 7. 25-45.

Mullet, K. and D. Sano (1995). Designing Visual Interfaces. Sun Microsystems, Inc. Mountainview, California.

Power, W.T., (1973). Behavior: The Control of Perception. Aldine de Gruyter, Inc. Hawthorne, New York.

Sinaiko, H. W. and E. P. Buckley (1957). Human Factors in the Design of Systems. NRL Report 4996, Naval Research Laboratory, Washington, D.C., August 1957. In Selected Papers on Human Factors in The Design and Use of Control Systems, Edited by H. W. Sinaiko, Dover Publications, Inc. New York, 1961.

Taylor, M. M. (1993). Principles for intelligent human-computer interaction: A tutorial on layered protocol theory (DCIEM Report No. 93-32). North York, Ontario: Department of National Defence.

Taylor, M. M. and D. A. Waugh (1991). Dialogue analysis using Layered Protocols. In W.J.Black & H.C.Bunt (Eds), Abduction, Belief, and Context. Studies in Computational Pragmatics, University College London Press, London, (in press).

Taylor M.M., P.S.E. Farrell, M.A.H. Semprie, & H.D., Hollands, J.G., (1997). Perceptual Control and Layered Protocols in Interface Design. II. The General Protocol Grammar. (in progress).

Van de Vegte, J., (1990). Feedback Control Systems (2nd Ed.). Prentice Hall, Englewood Cliffs, New Jersey.

Appendix A

GPG Definitions

GPG Definitions

A protocol node is the basic structure for illustrating the communication between two partners. The protocol node describes the control of a belief state defined by three propositions. If the current belief state does not match the reference belief state then the protocol node output would be some virtual message (form of feedback) that is transmitted to the partner. Eventually the partner's actions (or their virtual messages) become input to the protocol node that moves the current belief, hopefully, towards the reference belief.

The reference belief state for cooperative communication is that the recipient of a Primal Message has made an adequate interpretation and it is no longer worth continuing to transmit the message. This reference belief may be divided into three propositions (Taylor et al. 1997):
P1: The recipient has made or is in the process of making an interpretation of the primal message.
P2: The quality of the communication mechanism is sufficient for an adequate interpretation of a message.
P3: It is not worth continuing to improve the recipient's interpretation of this message.
Once there is a strong current belief in P1, P2, and P3, then one could say that the current belief matches the reference belief.

Normally, each partner may have some level of belief for each proposition. The following ordinal scale is defined to facilitate a short-hand description of the forms of feedback:

Strong Disbelief	= -1
Weak Disbelief	= wd (somewhere between -1 and 0)
No Opinion	= 0
Weak Belief	= wb (somewhere between 0 and 1)
Strong Belief	= 1

Also, we define a short-hand for the originator's belief in a proposition as O(p), and R(p) for the recipient. Therefore $-1 < O(P2) < 0$ means that the originator's belief that *the quality of the communication mechanism is sufficient for an adequate interpretation of a message* ranges from Strong Disbelief to No Opinion. Note that, this is different from $O(P2) = \{-1, wd, 0\}$ where a particular form of feedback is associated with each element within the list. Using this taxonomy, the reference belief state is $O(P1) = 1$, $O(P2) = 1$, $O(P3) = 1$, $R(P1) = 1$, $R(P2) = 1$, and $R(P3) = 1$.

The General Protocol Grammar (GPG) has nodes and arcs is illustrated in Figure A-1. The nodes represent plausible and current belief states of the originator that may exist during the transmission and interpretation of a message. For example, at OS it is likely that $O(P1) = -1$, $-1 < O(P2) < 1$, $O(P3) = -1$. In words, the originator believes the recipient has no adequate interpretation and it's worth continuing the communication. Table A-1 lists all nodes, their *mathematical* representation, and their meaning in words.

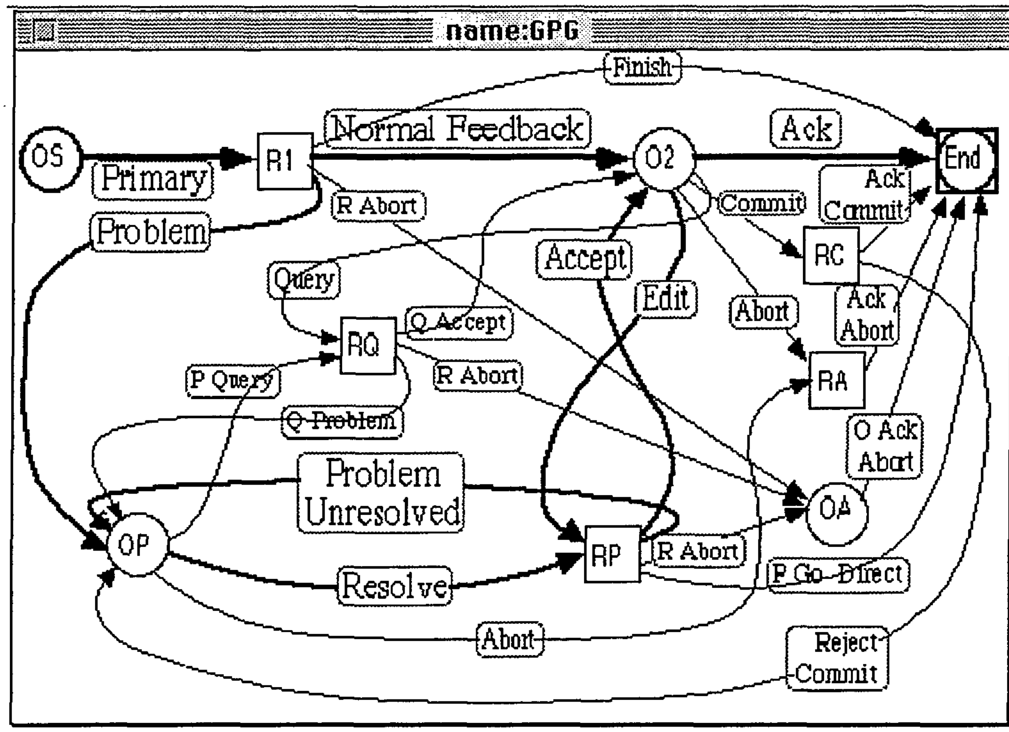


Figure A-1: Pictorial Representation of General Protocol Grammar

Table A-1. Description of GPG nodes.

Node	Current Beliefs	In words
OS	O(P1) = -1 -1 < O(P2) <1, O(P3) = -1	O believes R has no interpretation and it's worth continuing
O2	O(P1) = 1 0 < O(P2) < 1, O(P3) = -1	O believes R has an adequate interpretation and it's worth continuing
OP	O(P1) = 1 -1 < O(P2) < 0, O(P3) = -1	O believes R has an inadequate interpretation but it's worth continuing
OA	O(P1) = 1 O(P2) = -1, -1<O(P3) < 1	O believes R has an inadequate interpretation and it's not worth continuing
R1	O(P1) = 1 -1 < O(P2)<1, O(P3) = -1	O believes R has some interpretation and it's worth continuing
RP	O(P1) = 1 -1 < O(P2)<1, O(P3) = -1	O believes R has some interpretation and it's worth continuing
RC	O(P1) = 1 wb < O(P2) <1, O(P3) = 1	O believes R has an adequate interpretation and it's not worth continuing
RA	O(P1) =1 -1 < O(P2) < wd, O(P3) = 1	O believes R has an inadequate interpretation but it's not worth continuing
RQ	O(P1) =1 wd < O(P2)<wb, O(P3) = -1	O believes R has some interpretation and it's worth continuing
END *reference state*	O(P1) = 1 O(P2) = 1 O(P3) = 1	O believes R has an adequate interpretation and it's not worth continuing

The arcs represent the required form of feedback determined by the belief state of the receiver of a virtual message (either the originator or the recipient). For example, for the Primary arc, it is likely that O(R(P1)) = {-1,wd,0,wb,1}, -1 < O(R(P2)) < 1, O(R(P3)) = -1. In words, the originator (the sender in this case) believes the recipient (the receiver in this case) also believes that the recipient has a belief about the process of making an interpretation as denoted by {-1,wd,0,wb,1}. The originator might use *Inform* feedback if O(R(P1)) = {-1,wd,0}, *Neutral* feedback if O(R(P1)) = wb, or *Null* feedback if O(R(P1)) = 1 (see page 14, for complete descriptions of the forms of feedback).

If the recipient is now the sender then the chosen form of feedback would be based on the originator's belief state as receiver. For example, the Normal Feedback arc is denoted as follows: O(R(O(P1))) = 1, 0 < O(R(O(P2)) < 1, -1 < O(R(O(P3))) < 0). To simplify the reading of the mathematical expression, the recursive notation is dropped. Therefore, R(p) $\equiv$ O(R(p)), and O(p) $\equiv$ O(R(O(p))) when referring to feedback forms only. The following table lists all nodes and their meaning.

Table A-2. Description of GPG arcs and Feedback forms.

Arc	**Feedback Form**	**Current Beliefs**	**In words**
Primary		O(P1) = -1 O(P2) = 0 O(P3) = -1	O believes R has no initial interpretation and wants to begin transmission
	Null	R(P1) = 1 R(P2) = 0 R(P3) = -1	No feedback is required for an interpretation
	Neutral	R(P1) = {wb,0} R(P2) = 0 R(P3) = -1	Some feedback is required for an interpretation
	Inform	R(P1) = {-1,wd,0} R(P2) = 0 R(P3) = -1	Inform feedback is required for an interpretation
Normal Feedback *or* Accept		R(P1) = 1 0 < R(P2) <1 R(P3) = -1	R believes R has some adequate interpretation and wants to communicate this to O
	Null	O(P1) = 1 O(P2) = 1 O(P3) = {-1,wd,0}	No feedback is expected for an adequate interpretation
	Neutral	O(P1) = 1 O(P2) = {wb,0} O(P3) = {-1,wd,0}	Some feedback is expected for some adequate interpretation
	Inform	O(P1) = 1 O(P2) = {wb,0} O(P3) = {-1,wd,0}	Inform feedback is expected for an adequate interpretation

N.B. some of the arcs (and feedback forms) have identical mathematical expressions. In these cases, the arc's uniqueness depends on the trend of the message (i.e., what came before it and what is expected to follow).

Arc	Feedback Form	Current Beliefs	In words
Problem *or* Problem Un-resolve		R(P1) = 1 -1 < R(P2) < wd R(P3) = -1	R believes R has some inadequate interpretation and wants to communicate this to O
	Neutral	O(P1) = 1 O(P2) = wd O(P3) = 0	Some feedback is expected for some adequate interpretation
	Inform	O(P1) = 1 O(P2) = {wd,-1} O(P3) = {-1,wd,0}	Inform feedback is expected for some adequate interpretation
Finish *or* P go Direct		R(P1) = 1 R(P2) = 1 R(P3) = 1	R believes R has an adequate interpretation and wants to end the communication
	Null	O(P1) = 1 O(P2) = 1 O(P3) = 1	No feedback is expected for an adequate interpretation
	Neutral	O(P1) = 1 O(P2) = wb O(P3) = 1	Some feedback is expected for some adequate interpretation
R Abort		R(P1) = 1 R(P2) = {-1,wd,0} R(P3) = 1	R believes R has an inadequate interpretation and wants to end the communication
	Neutral	O(P1) = 1 O(P2) = 0 O(P3) = {0,wb,1}	Some feedback is expected that it's not worth continuing
Acknow-ledge		O(P1) = 1 O(P2) = {wb,1} O(P3) = 1	O believes R has an adequate interpretation and wants to end the communication
	Null	R(P1) = 1 R(P2) = 1 R(P3) = 1	No feedback is required for an adequate interpretation
	Neutral	R(P1) = 1 R(P2) = wb R(P3) = 1	Some feedback is required for an adequate interpretation
Commit		O(P1) = 1 O(P2) = 1 O(P3) = 1	O believes R has an adequate interpretation and wants to end the communication
	Neutral	R(P1) = 1 R(P2) = wd R(P3) = 0	Some feedback is required to end even though R believes R has some inadequate interpretation
	Inform	R(P1) = 1 R(P2) = {-1,wd} R(P3) = 1	Inform feedback is required to end even though R believes R has an inadequate interpretation

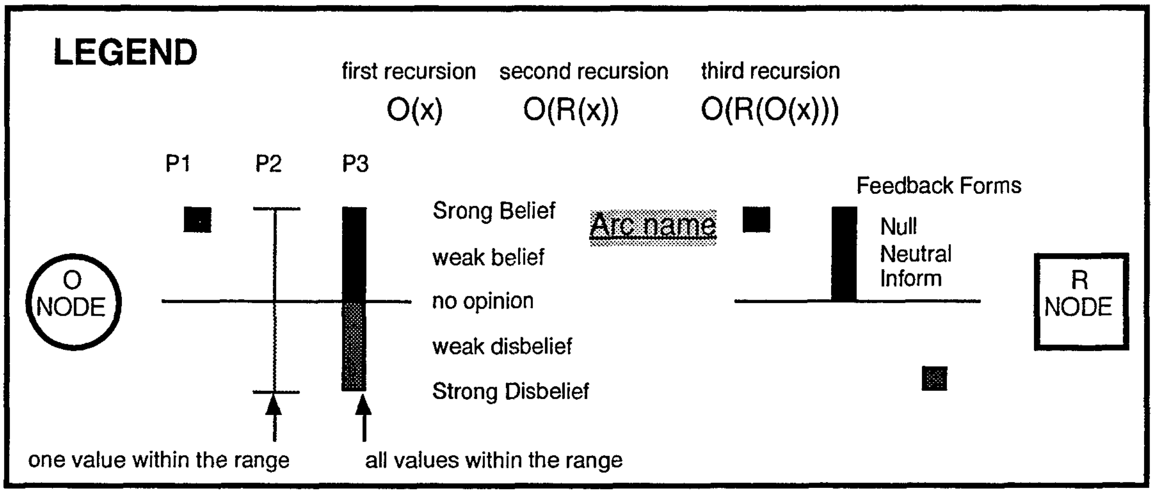
Arc	Feedback Form	Current Beliefs	In words
Abort		O(P1) = 1 -1 < O(P2) < 1 O(P3) = 1	O wants to end the communication regardless of the interpretation
	Null	R(P1) = 1 -1 < R(P2) < 1 R(P3) = 1	No feedback is required since R also has a strong opinion that it's not worth continuing
	Neutral	R(P1) = 1 -1 < R(P2) < 1 R(P3) = {wd,0,wb}	Some feedback is required since R has a weak opinion about whether to continue
	Inform	R(P1) = 1 -1 < R(P2) < 1 R(P3) = -1	Inform feedback is required since R believes has a strong opinion to continue
Edit *or* Resolve		O(P1) = 1 O(P2) = wb O(P3) = -1	O believes R has some interpretation and wants to make it adequate
	Neutral	R(P1) = 1 R(P2) = wb -1 < R(P3) <1	Some feedback is required since R has made some adequate interpretation
	Inform	R(P1) = 1 R(P2) = {0,wb} -1 < R(P3) <1	Inform feedback is required since R has made some adequate interpretation
Query		O(P1) = 1 O(P2) = 0 O(P3) = -1	O believes R has an interpretation but clarification is required
	Neutral	R(P1) = 1 R(P2) = {wb,1} -1 < R(P3) <1	Some feedback is required since R has made an adequate interpretation
	Inform	R(P1) = 1 R(P2) = {0,wb} -1 < R(P3) <1	Inform feedback is required since R has made some adequate interpretation
P Query		O(P1) = 1 O(P2) = 0 O(P3) = -1	O believes R has an interpretation but clarification is required
	Neutral	R(P1) = 1 R(P2) = {wd,0} -1 < R(P3) <1	Some feedback is required since R has made some inadequate interpretation
	Inform	R(P1) = 1 R(P2) = {-1,wd} -1 < R(P3) <1	Inform feedback is required since R has made an adequate interpretation
Reject Abort		O(P1) = 1 -1 < O(P2) < 1 O(P3) = -1	O believes R wants to prematurely end and wants to continue the communication
	Neutral	R(P1) = 1 -1 < R(P2) < 1 R(P3) = wb	Some feedback is required in order for the message to continue
	Inform	R(P1) = 1 -1 < R(P2) < 1 R(P3) = 0	Inform feedback is required in order for the message to continue

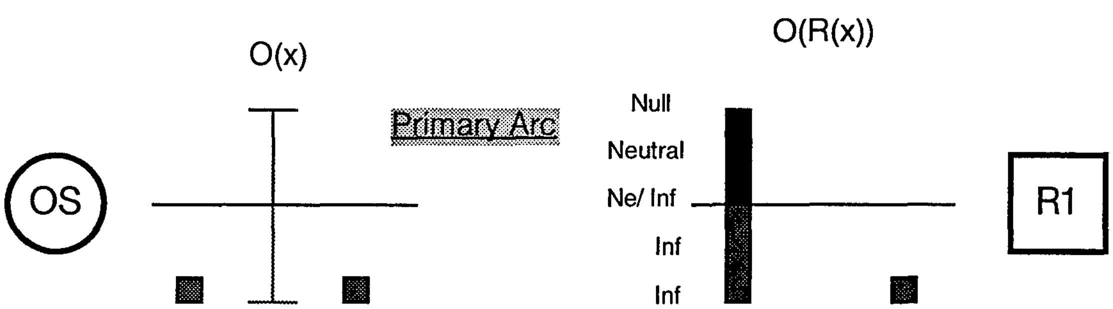
Arc	Feedback Form	Current Beliefs	In words
Ack Commit		R(P1) = 1 R(P2) = {wb,1} R(P3) = {wb,1}	R believes R has an adequate interpretation and acknowledges O's desire to end the message
	Neutral	O(P1) = 1 -1< O(P2) < 1 O(P3) = {wb,1}	Some feedback is expected to end the message
	Inform	O(P1) = 1 -1< O(P2) < 1 O(P3) = wb	Inform feedback is expected to end the message
Reject Commit		R(P1) = 1 R(P2) = {-1,wd} R(P3) = {-1,wd}	R believes R has an inadequate interpretation and does not want to end the message
	Inform	O(P1) = 1 -1< O(P2) < 1 O(P3) = {-1,wd}	Inform feedback is expected to continue the message
Ack Abort		R(P1) = 1 R(P2) = 0 R(P3) = 1	R believes R has an interpretation but agrees to end the communication
	Null	O(P1) = 1 -1< O(P2) < 1 O(P3) = 1	No feedback is expected to end the message
	Neutral	O(P1) = 1 -1< O(P2) < 1 O(P3) = wb	Some feedback is expected to end the message
Q Accept		R(P1) = 1 R(P2) = {0,wb} R(P3) = -1	R believes R has an adequate and wants to answer O's query
	Inform	O(P1) = 1 O(P2) = 0 O(P3) = -1	Inform feedback is expected since the adequacy of the interpretation is unknown
Q Problem		R(P1) = 1 R(P2) = {wd,0} R(P3) = -1	R believes R has an inadequate and wants to answer O's query
	Inform	O(P1) = 1 O(P2) = 0 O(P3) = -1	Inform feedback is expected since the adequacy of the interpretation is unknown

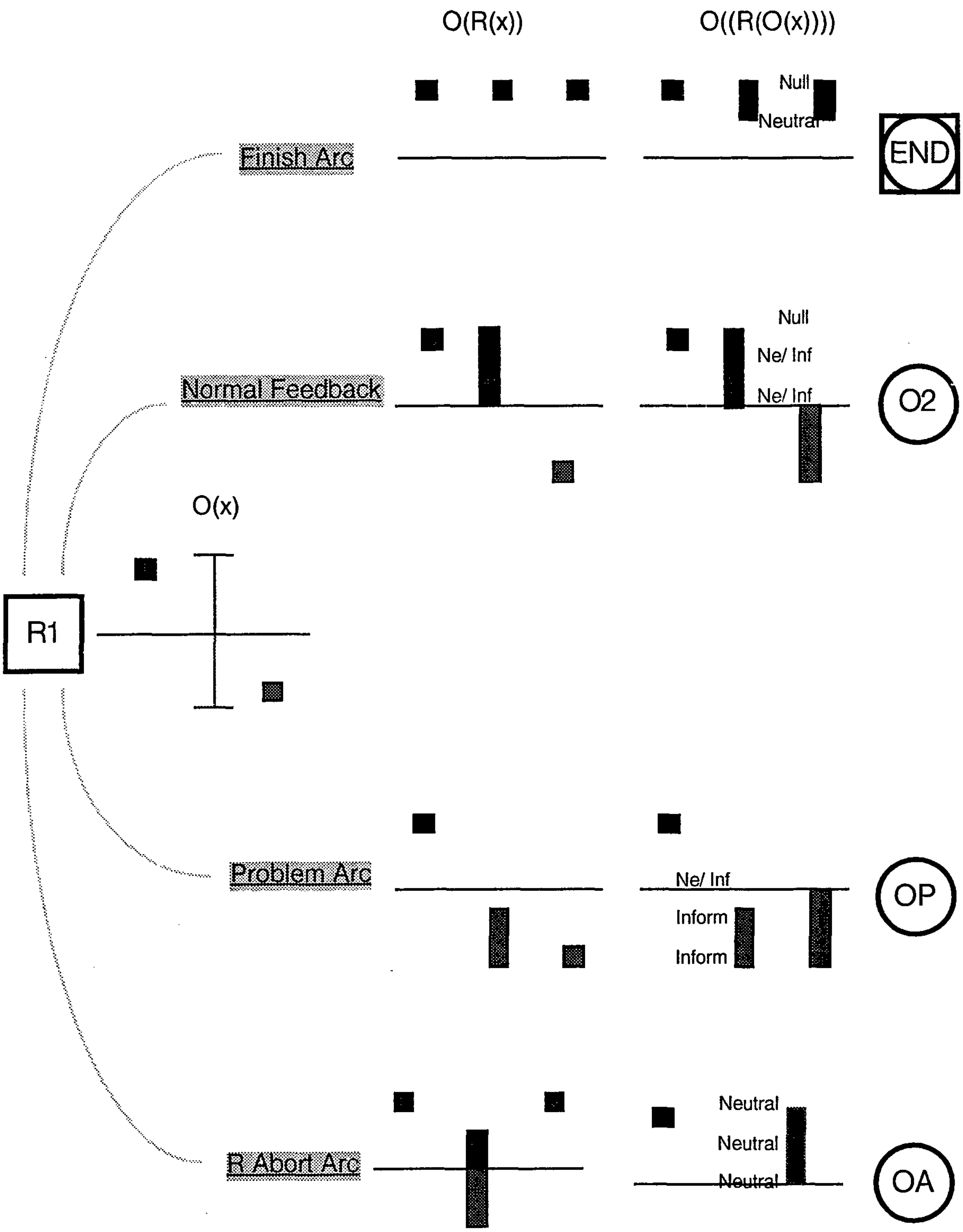
The table reveals very slight differences in formulating the nodes, arcs, and feedback forms. In some cases, the differences depend on the trend of the conversation. In other instances, the intensity of the belief is the only thing that separates the appropriate form of feedback. However, the GPG is general enough to cover most human-human communication. These descriptions of the nodes and arcs become a good reference for the following Appendices.

The GPG descriptions are shown below in pictorial form in order to see the connectivity and flow of belief states from one node, across an arc, to another. This is not to show that beliefs change smoothly and in a predictable

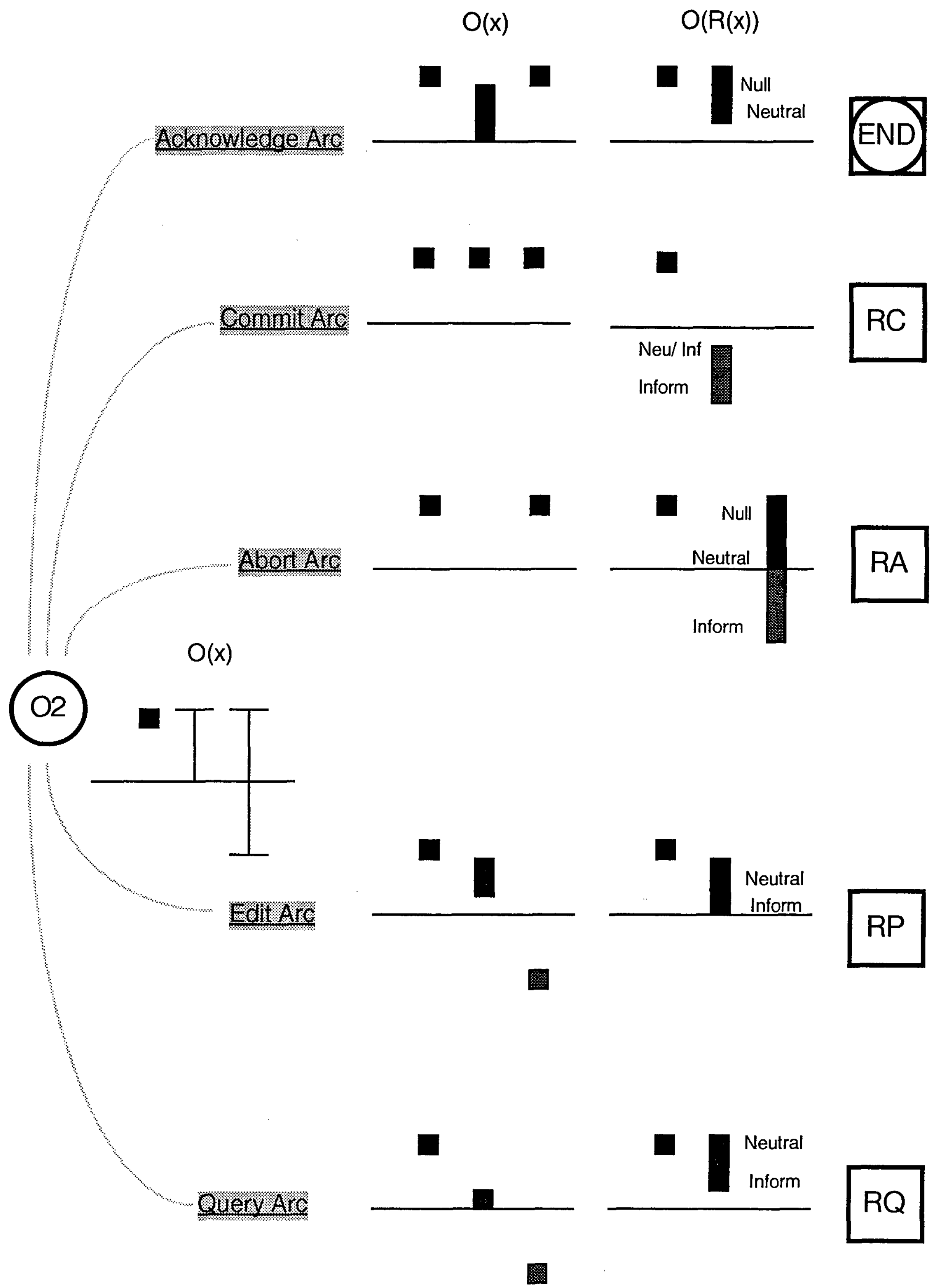
manner. In fact, one could imagine that a bar graph exists at each and every node and arc within the GPG simultaneously. As the conversation proceeds, the bar graph levels fluctuate according to the partners' belief states. Conflicts may occur such as the originator being at the end node while the recipient is still at the primary node. However, in most cases, one could imagine a wave of beliefs that move the bar graphs representing the three propositions starting from the Primary node and strong disbelief (red) to the End node and strong belief (blue).

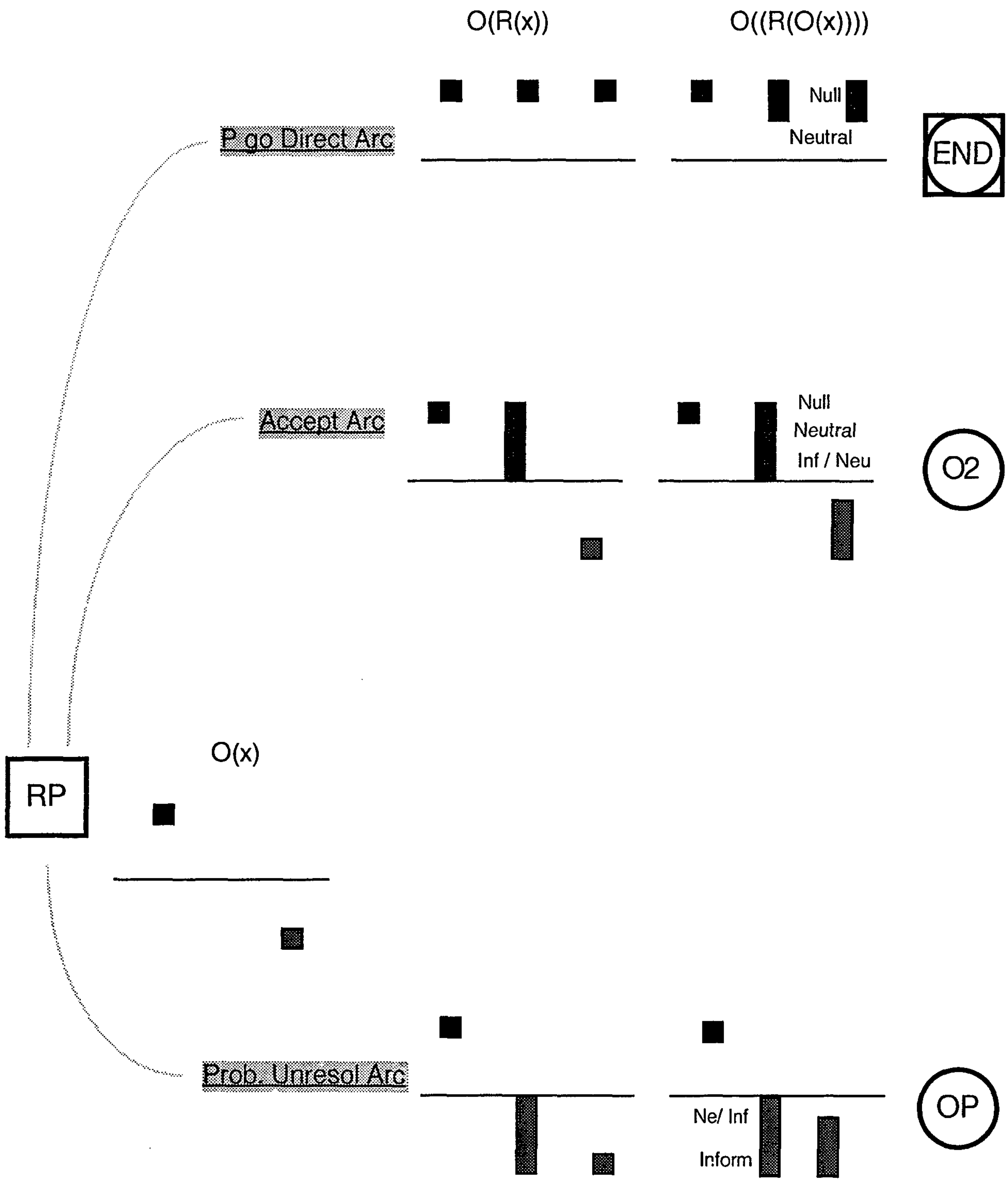


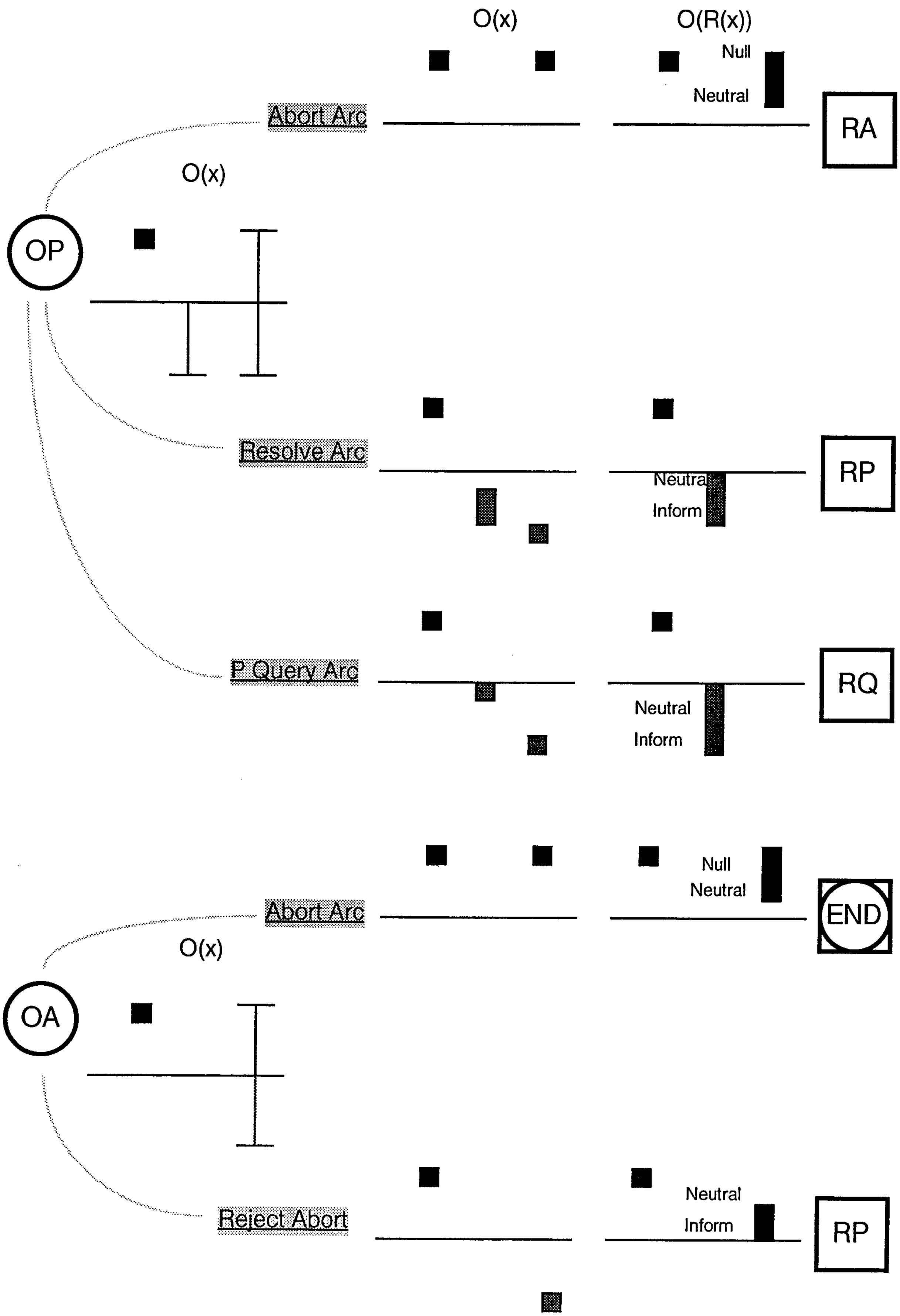


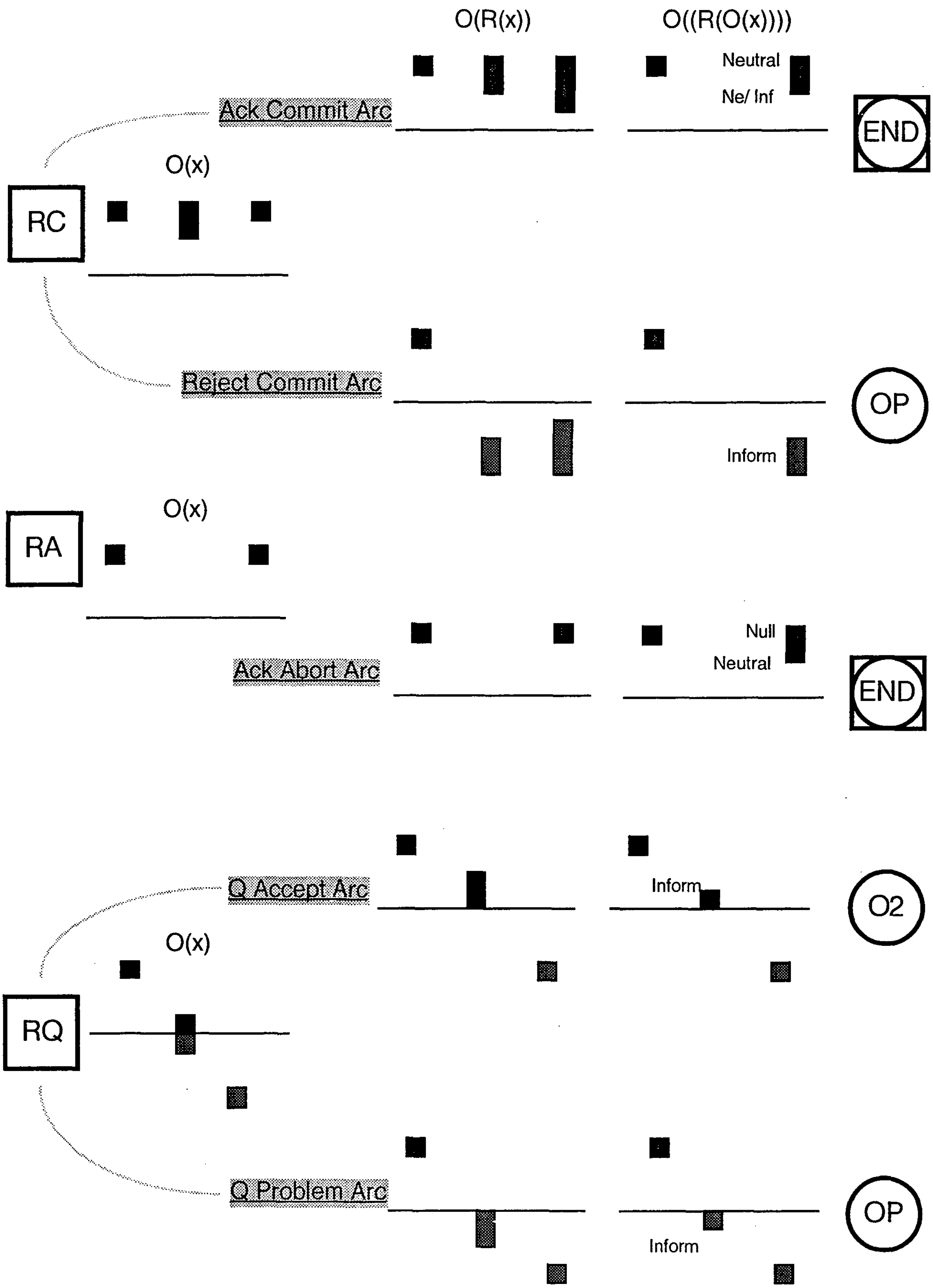


O(R(x))
O((R(O(x))))
Null
Neutral
Finish Arc
END
Null
Ne/ Inf
Ne/ Inf
Normal Feedback
O2
O(x)
R1
Problem Arc
Ne/ Inf
Inform
Inform
OP
Neutral
Neutral
Neutral
R Abort Arc
OA


O(x)
O(R(x))
Acknowledge Arc
Null
Neutral
END
Commit Arc
Neu/ Inf
Inform
RC
Abort Arc
Null
Neutral
Inform
RA
O2
O(x)
Edit Arc
Neutral
Inform
RP
Query Arc
Neutral
Inform
RQ


O(R(x))
O((R(O(x))))
P go Direct Arc
Null
Neutral
END
Accept Arc
Null
Neutral
Inf / Neu
O2
O(x)
RP
Prob. Unresol Arc
Ne/ Inf
Inform
OP


O(x)
O(R(x))
Null
Neutral
Abort Arc
RA
O(x)
OP
Resolve Arc
Neutral
Inform
RP
P Query Arc
Neutral
Inform
RQ
Null
Neutral
Abort Arc
END
O(x)
OA
Reject Abort
Neutral
Inform
RP



O(R(x))
O((R(O(x))))
Neutral
Ne/ Inf
Ack Commit Arc
END
O(x)
RC
Reject Commit Arc
OP
Inform
O(x)
RA
Null
Neutral
Ack Abort Arc
END
Q Accept Arc
Inform
O2
O(x)
RQ
Q Problem Arc
Inform
OP

Appendix B

Listing of Network Views

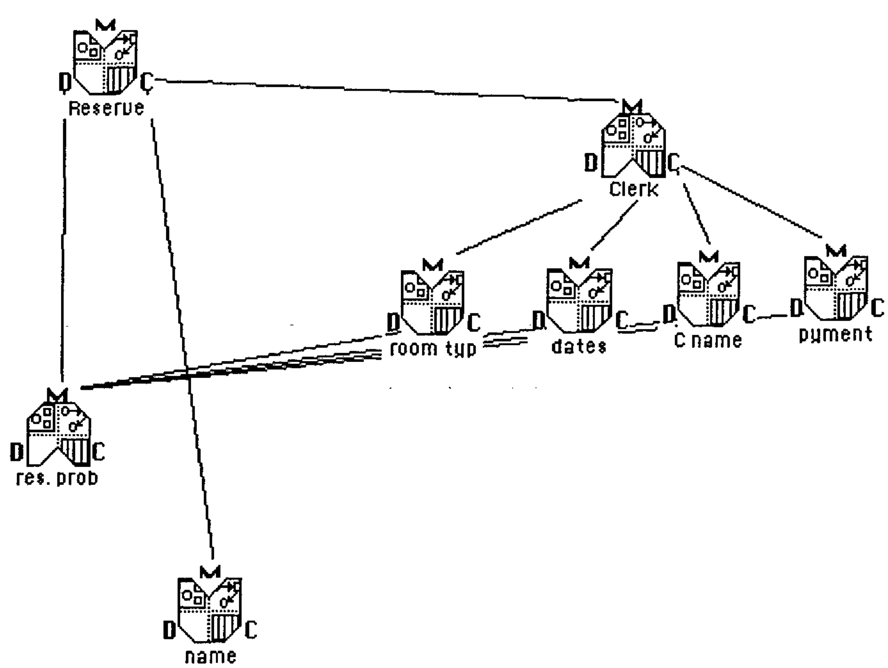
Listing of Network Views

This following is a of the LPTool Network View models listing with short descriptions that were generated during the initial stages of the analysis. The models are presented in chronological order.

Client_Hotel Network View

The Client_Hotel model was originally constructed in order to help us understand the nuances of cooperative communication. The scenario is common. For example, a patron (transmitting node) enters a hotel with a desire to rent a room and the clerk (receiving node) behind the desk wants to rent out the rooms of the hotel. In order for either Primary message to be satisfied, the clerk must get particular information from the client and the client must give this information readily.

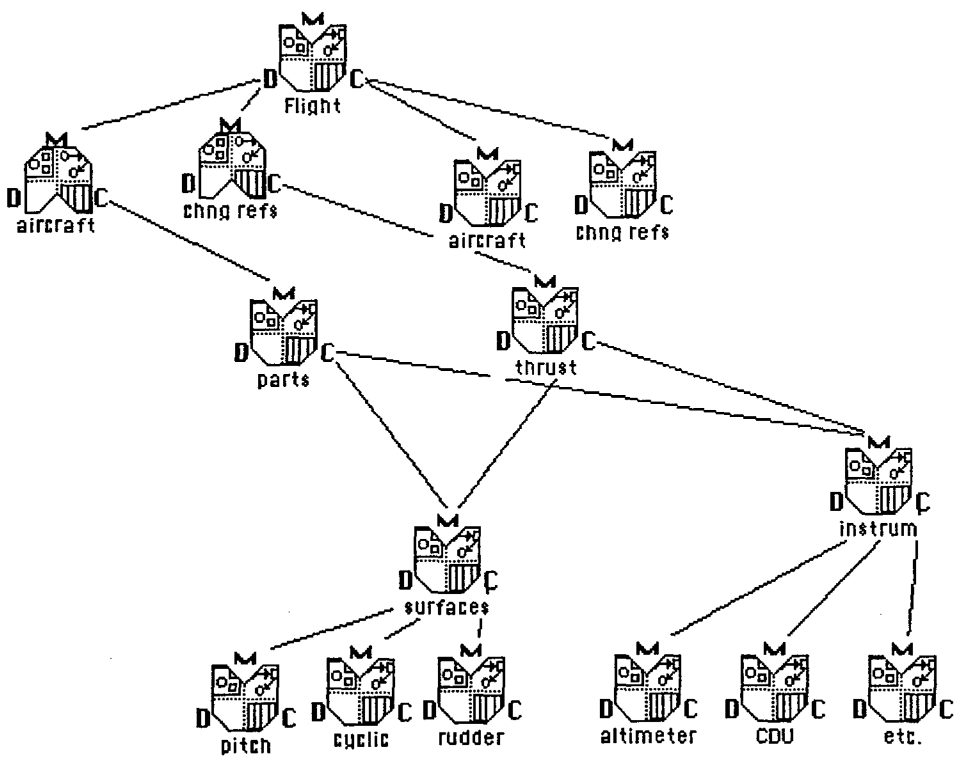
In retrospect, the client_hotel model is an inaccurate portrayal of this type of interaction. Both the client's and the clerk's Network views are mapped onto the same D-model. This model was created before an understanding of the usage of transmitting nodes and receiving nodes were obtained. It is not possible to have a receiving node come directly from the Coder. Finally, the group of protocols stemming from the clerk's Coder should be grouped in one protocol node labelled 'client information'.



Marc Network View

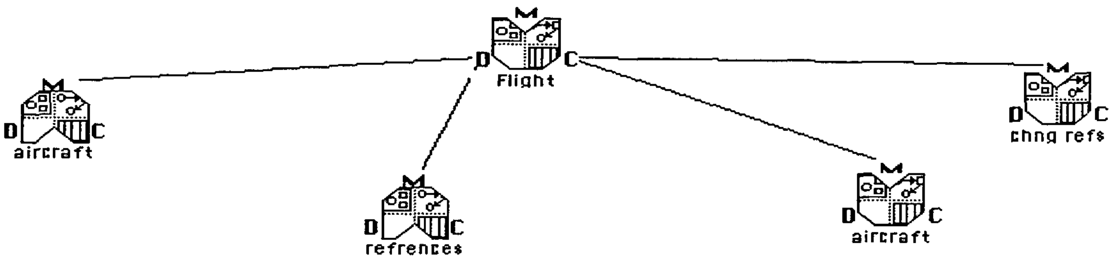
The model Marc was constructed as an attempt to see what the human pilot must have in order to perceive flight. The emphasis in this model is on the transmitting nodes and the motions that must be completed in order to believe flight is occurring.

'Marc' shows a systematic breakdown of the elements involved in the flying process. At this point, it was thought that a protocol was an individual concept. Therefore, an interaction could be described by labelling the main concepts that enable a perception. For example, in order to achieve flight an aircraft is needed as well as a change in reference points, i.e., instruments, ground, clouds, etc.. In order to have an aircraft, parts are needed and some form of thrust is needed in order to have changing references.



Marc2 Network view

The first attempt, Marc1, was lost due to an application error. Marc2 was the second attempt to integrate both the Decoder and Coder into the same model. Although the protocol nodes have the same titles, they hold different GPG's. The GPGs on the Decoder side of Flight describe the elements that must be perceived in order for a pilot to believe an aircraft and changing references are present. The GPGs on the Coder side of Flight describe what must be done, or actions to be completed, in order to obtain the perception.



Flight Model Network View

The Flight model was derived by asking the question, "Assuming that the pilot is sitting in a cockpit, what are the things that will give a pilot a perception of flight?" The simple answer to this question was information gathered through the senses.

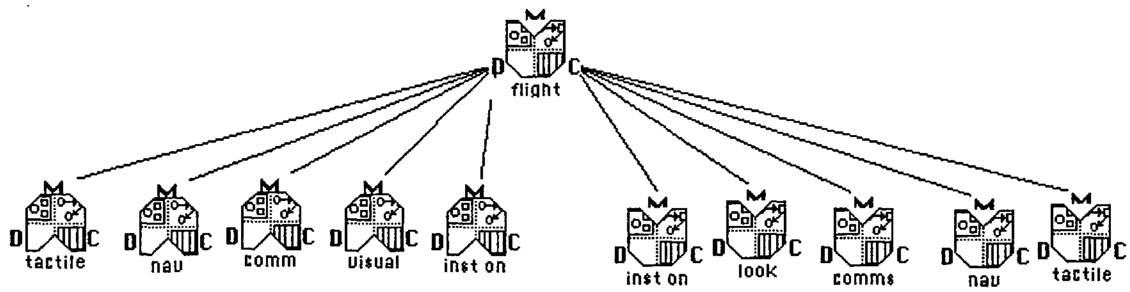
From this, it was gathered that some sort of tactile perception e.g., the buoyancy felt when in the air, or turbulence, was one factor that is sufficient to inform a pilot that they are flying.

The act of navigation is also something that can inform a pilot of flight. The constant changing of external references will be sufficient to give the pilot the perception of motion. When combined with the tactile perception, the perception of flight is either accepted or rejected.

Communication from an external source e.g., control tower, might also be sufficient to give the pilot a perception of flight. The pilot may not be able to derive enough information from their own devices and therefore be required to use the perceptions of others to reinforce their own perceptions.

Visual cues will also give the pilot an indication of what the status of the aircraft.
Finally, the instruments in an aircraft are sufficient enough to give the perception of flight. This is the principle on which some flight simulators are built.

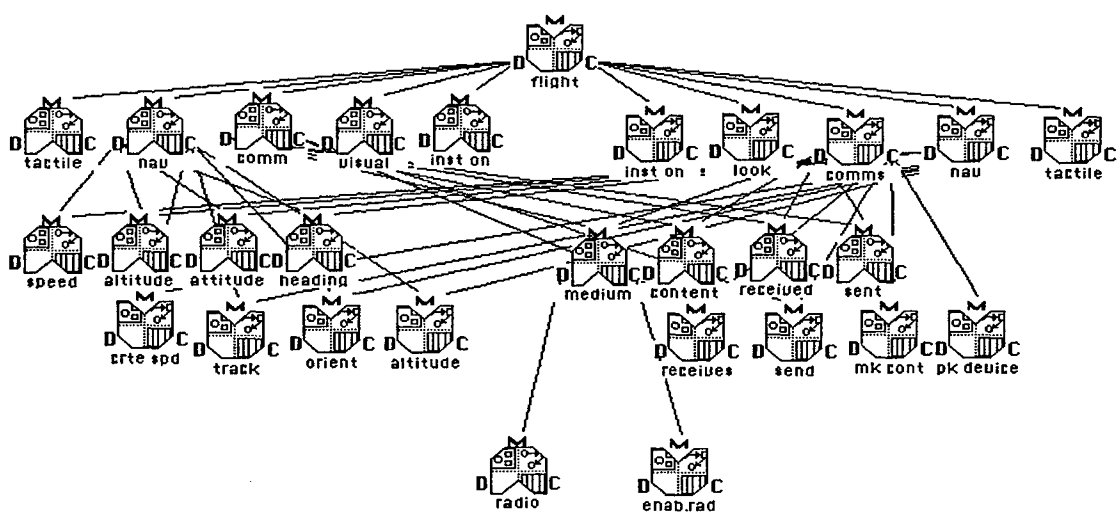
Flight Model Network View



Flight 2b Model Network View

Flight 2b was an extension to the Flight model that began to examine the elements that compose navigation and communication. From the pilot's point of view, the elements composing navigation are speed, altitude, attitude and direction. With this information the pilot is able to derive vectors and navigate accordingly.

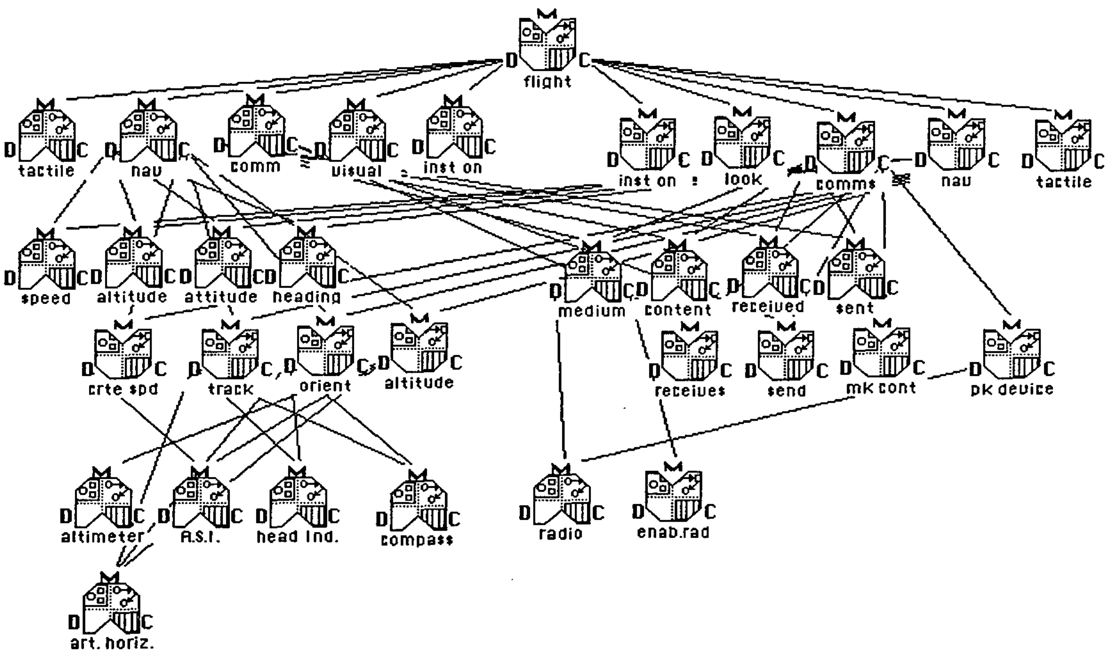
Again, considering the pilot's point of view regarding communication, some sort of medium in which to communicate is mandatory and content of communiqué as well as communiqué being safely sent and received, will complete the communication loop.



Flight 3a Model Network View

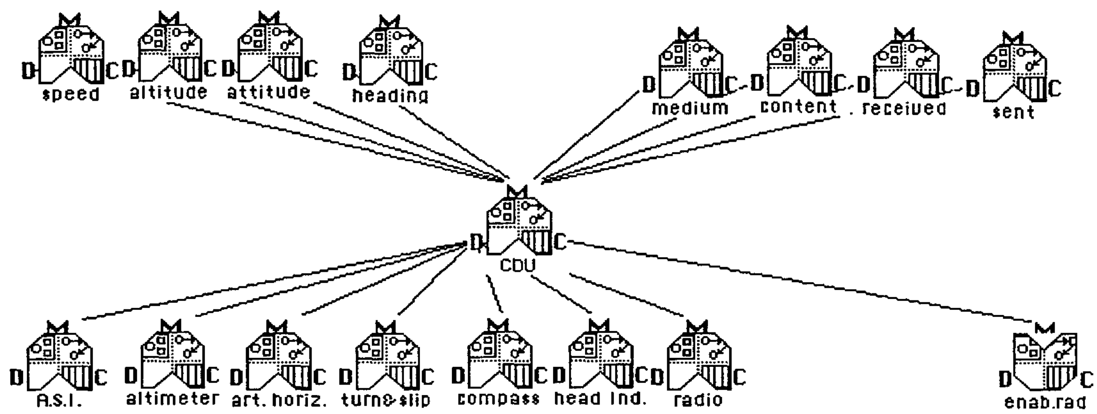
The Flight 3a model attempts to link the desired perceptions to a set of displays that are more efficient and understandable to the pilot. Confusion arose as to where the instruments, i.e., altimeter, heading indicator, airspeed indicator, artificial horizon and compass, were to fit into the scheme of things.

The Flight 3a model was the first practice model that had "real" implications for the development of the CDU. It was hypothesised that if the CDU was in fact an important apparatus, then its existence will naturally fall out of the GPG of a higher protocol node. The GPG's of the instruments were filled out either haphazardly, or, not at all, therefore, the CDU did not ensue.



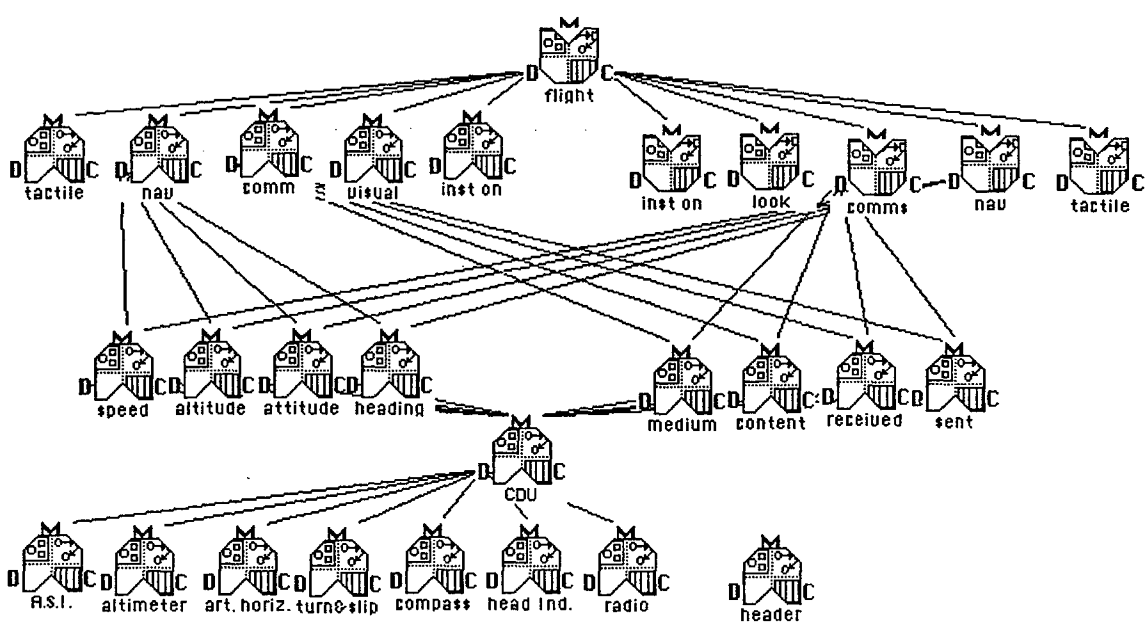
Comm/Nav Model Network View

The Comm/Nav model takes the analysis back a few steps to examine the GPG's of the appropriate nodes in order to figure out if the CDU will naturally fall out of the analysis. The CDU does in fact come out in the design, however, this model skips many necessary steps. After the level of CDU, the model jumps to a near final level of abstraction



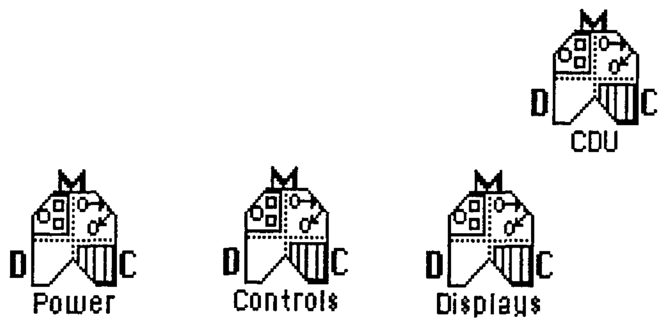
Flight 3c Model Network View

The Flight 3c Model is an amalgamation of the two previous models with no other changes added. This view was added so that the complete picture of the D-model could be observed at once.



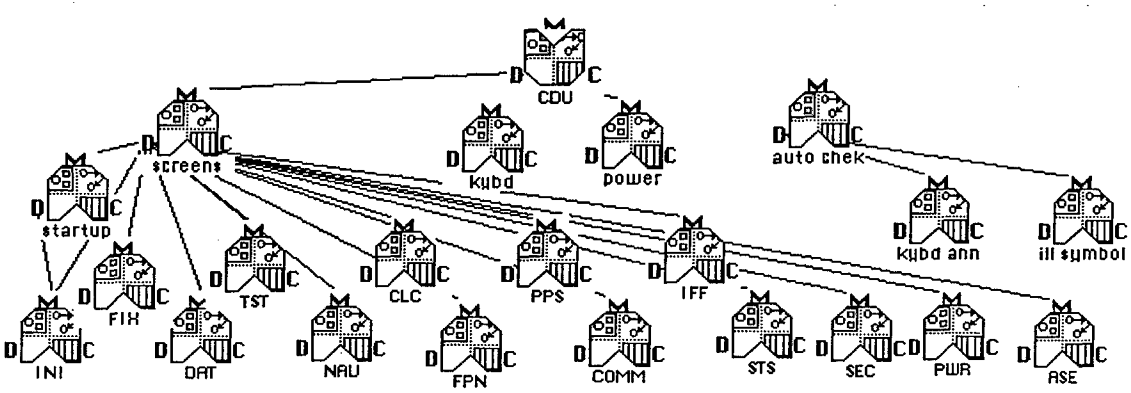
Paper CDU Model Network View

The Paper CDU model was an attempt to streamline the CDU D-model into the most basic of elements, i.e., controls and displays. This was the first time that the notion of general protocols were touched upon. This model was altered shortly after this for two reasons; firstly, the protocols were in fact too general. Secondly, the controls and displays resided on the last (or lowest) level of abstraction that was being examined, thus, skipping many intermediate steps.



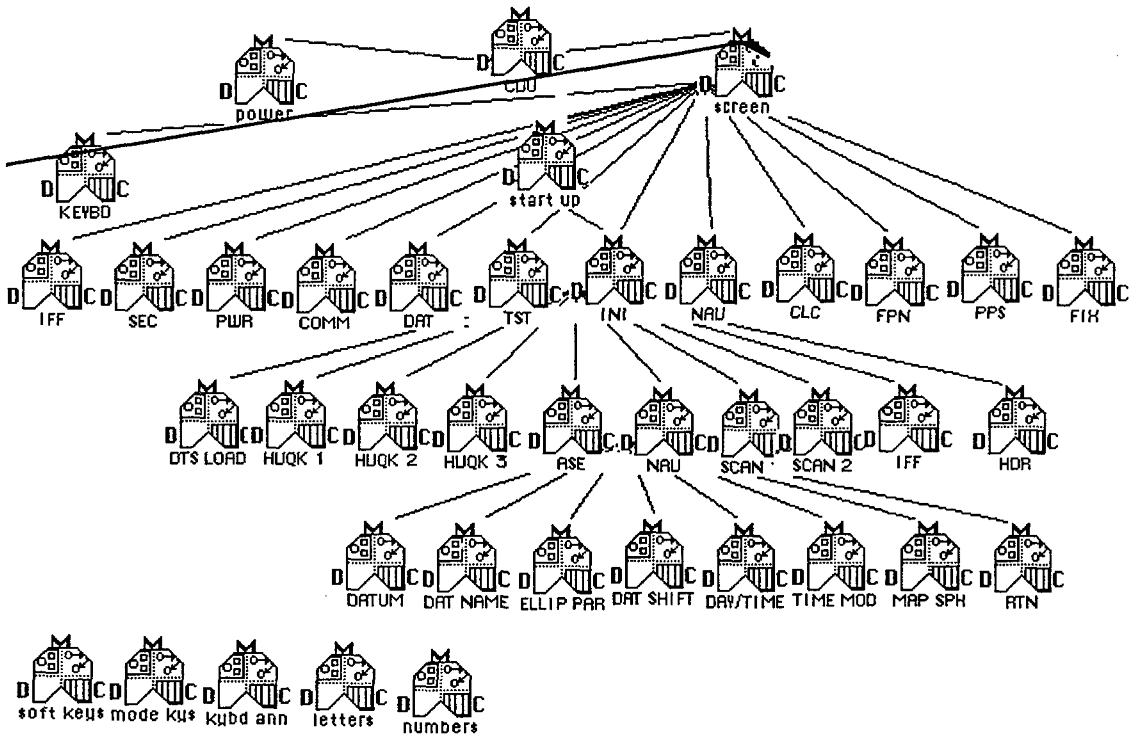
CDU Transmitter Model Network View

The CDU Transmitter model was designed to see what it would be like to view things from the transmitting or active (as opposed to passive) perspective. This model was derived with the aid of the CMC Operations Manual. CDU Transmitter was abandoned as quickly as it arose because it was not clear if the GPG of CDU had any of the following screens falling out of it naturally.



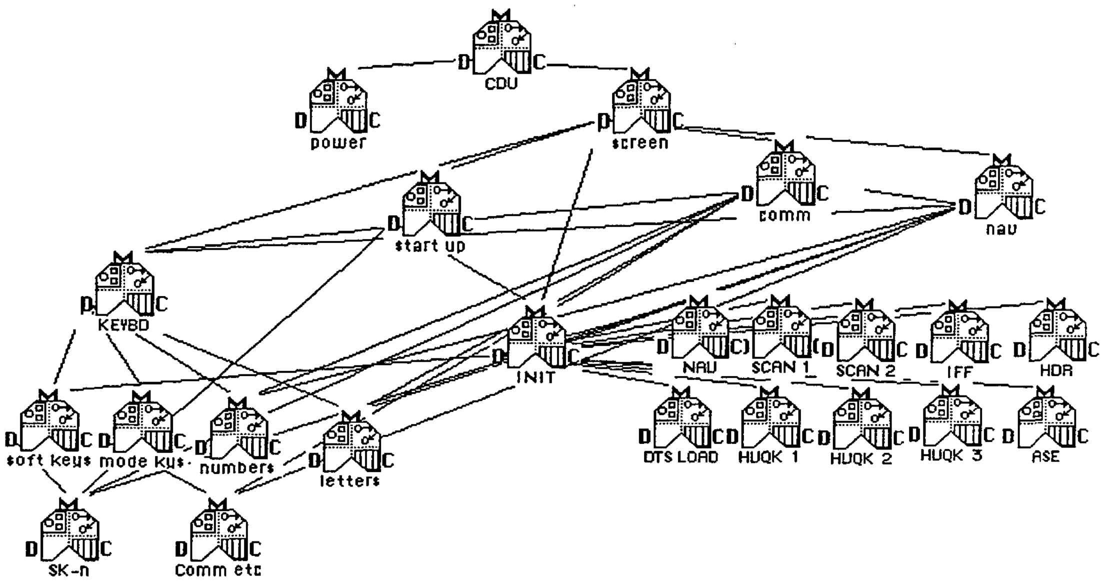
CDU VSD1 Model Network View

Carrying on from the example set in CDU Transmitter, the CMC Operations Manual was used to gain an idea into what an LPT analysis of the CMC CDU would look like. This model shows only as far as the initialisation procedures for the CDU.



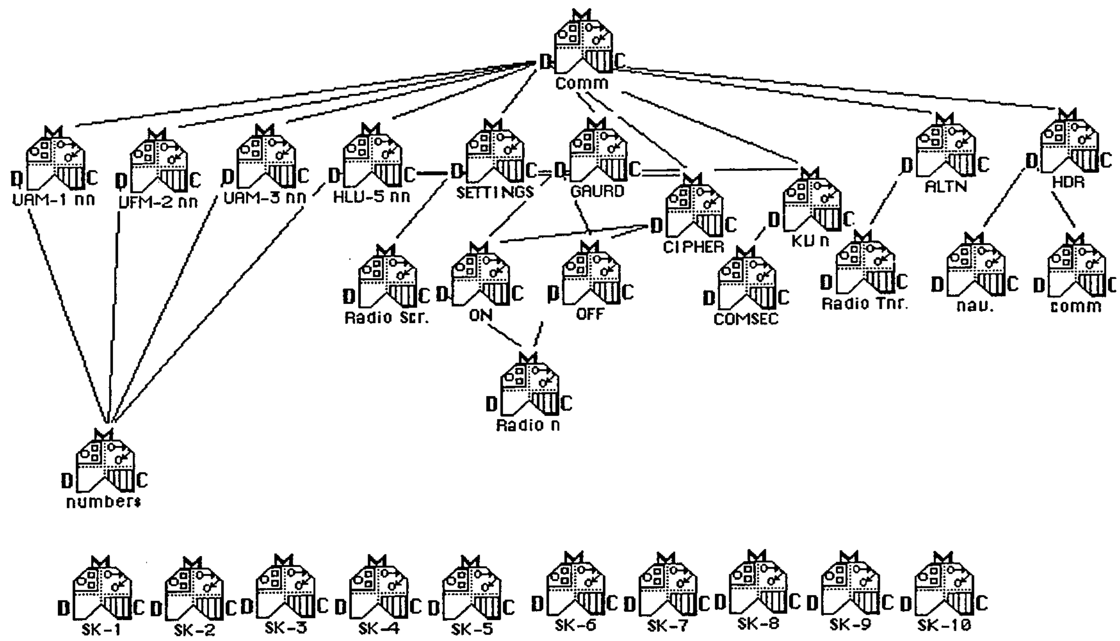
CDU Variable Screen Data 1 Model Network View

The CDU Variable Screen Data 1 Model contains fully instantiated CDU, Power, Screen, Start up and INIT nodes however the form of this model was taken from the CMC Operations Manual. This model may be misleading due to the combining of two models into one. The nodes coming from the Screens Protocol, with the exception of KEYBD and Start Up, should not actually be seen in this view. The Screen Protocol has in essence been magnified to show what is inside, similar to the KEYBD Protocol, which endures the same process.



Comm Model Network View

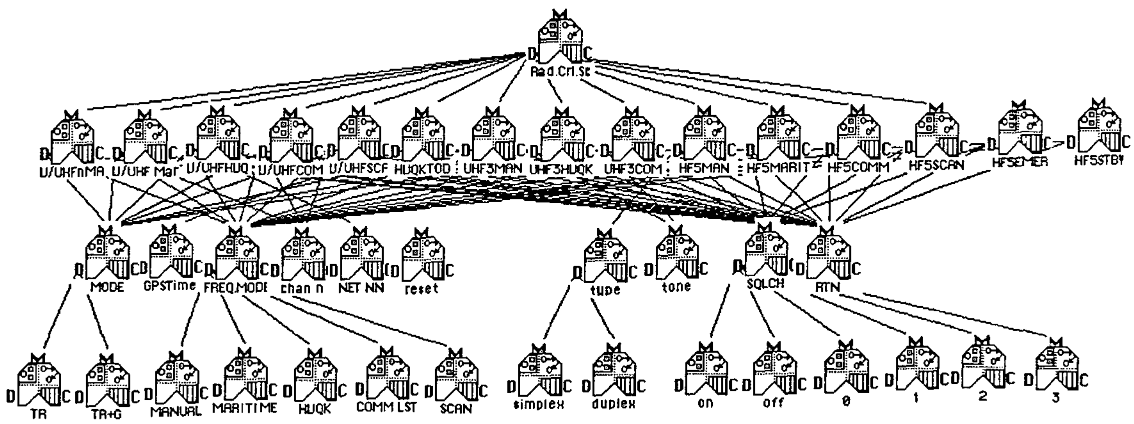
The Comm Model was an attempt to isolate one of the variables in the existing model to determine possible design flaws that might inhibit the full use of the CDU. The pilots must engage in a lengthy process in order to engage a radio. This is not only time consuming; it is also confusing to the user. The pilot will require a great deal of memory power in order to navigate successfully through the menu structures without getting lost. The second portion of this model is found in Radio Control Screens Model, where the Protocol, Radio Scr. in this model, has been expanded in order to map the entire radio enabling procedures.



Radio Control Screens Model Network View

The Radio Control Screens Model is a continuation of the Comm Model that was to use LPT to analyse the existing interactions of the Radio sequence. The CMC Operations Manual was used exclusively.

Through models such as this one, the deficiencies of the LPTool showed through. It is nearly impossible to make out where the diagram's connecting lines are coming from, thus increasing the difficulty in reading the interactions. This point is looked at more in Appendix D.



Appendix C

Full GPG for Pilot-CDU Interaction Model

Full GPG for Pilot-CDU Interaction Model

The following is a list of the annotations that appear in the General Protocol Grammar View in the LPTool Model of the pilot-CDU interaction.

Communication node

Arc	Form of Feedback	Annotation
E-feedback	Inform	The aircraft system informs the pilot that it is ready to begin to establish the link. If not ready, then there should be the appropriate PN's to support this arc such as: 1) power up aircraft system 2) CDU ready.
Primary	Inform	The pilot may choose to inform the CDU that the pilot wants see a link established. This may be done by depressing the "com" key on the CDU interface.
Normal Feedback	Neutral	Since the CDU is dual functional, it is highly unlikely that, at this point, the CDU will know the content of the message other than a message has been passed. The CDU can only respond by showing a start-up screen when the CDU is turned on.
	Verify	In the case where the pilot has depressed the COM Key the CDU responds by displaying a COMM SUMMARY screen. This is VERIFY feedback.
Acknowledge	Null	The pilot acknowledges the COMM SUMMARY screen by simply looking at it. The pilot does not need to pass a message to the CDU indicating that it has acknowledged the receipt of the message.
Abort	Neutral	Now, at any point the pilot may want to not establish a communications link. At the very least, all the pilot needs to do is turn off the power. However, an abort could occur by not satisfying the individual perceptions at lower levels which support this node.
Ack Abort	Neutral	The CDU will acknowledge the abort message by changing its state.

Navigation Node

Arc	Form of Feedback	Annotation
E-Feedback	Inform	The aircraft system will wait for the pilot to indicate that it is their wish to set waypoints. The system will be informed when the pilot pushes the Nav key. Things that support this perception are: 1) CDU 2) Nav Key
Primary	Neutral	The presence of a Nav key enables the CDU to send the message that it is ready to set waypoints. The pilot will acknowledge this message when the button is pressed. Things that support this perception are: 1) Nav Key
Normal Feedback	Neutral	The pilot wishes to set waypoints and therefore acknowledges the Navigational Primary message. The pilot may now press the nav key.
Acknowledge	Neutral	The CDU acknowledges the pilot's desire of setting waypoints by presenting the NAV screen. Things that support this perception are: 1) Screens

CDU Node

Arc	Form of Feedback	Annotation
Primary	Null	The CDU cannot initiate the Primary message in any specific way expect that it is there. This is one way of looking at it. Check out inform.
	Inform	The CDU informs the pilot that it has the potential to establish a communications link. This is done by having the COM, DAT, INI, STS, SEC, TST, POW, and startup keys visible and accessible to the pilot.
Finish	Null	The pilot may believe that the current radio link is in the desired state and simply ends the message with respect to establishing a radio link. The difference between the Null and Neutral forms of feedback are that, with Neutral, the pilot may be talking on the radio with the current link settings. With Null, the pilot does not make reference to the link either implicitly or explicitly.
	Neutral	same as Null
Normal Feedback	Null	The Null instantiation means that both partners have a strong belief about the transmission and content of the Primary message. In other words, the pilot may realise that the CDU wants to assist, but may not want its assistance right now.
	Neutral	This is difficult to imagine that the pilot would say to the CDU, "OK, I belief that you want to establish a radio link with me!" This is more of a NULL instantiation. However, by turning on the machine, etc. one can imagine that Neutral Feedback is given. Therefore the "startup" node supports this instantiation.
	Verify	The pilot may inform the CDU that the pilot is ready for its assistance. The pilot may do so by sending Virtual messages about the communication components such as: 0) starting up the CDU 1) initialising the comm settings 2) inputting the comm data 3) checking the communication system status 4) testing the communication system status 5) setting the security protocol 6) powering up or down the radios. Each one of these messages are implemented as the pilot interacts with the screens and keyboard. However, all of these messages together means that a link is being established.
Problem	Propose	Note that there may be a problem with the CDU's Primary message, in that it wants to perceive a communications link established. However, the CDU may be currently off, or in a different mode other than communication. The pilot then proposes the proper primal message by transmitting Virtual messages related to the communication components: in particular: 0) startup 1) Initialisation 2) comm summary 3) system status 4) testing 5) secure channel 6) power management
R Abort	Neutral	The pilot may initiate an abort if the pilot does not want to continue with this message of assistance in communicating. The instantiation of this message might be to turn off the CDU, or switch to another mode

Resolve	Inform	The CDU transmits the new Primary message by displaying the related screens.
Problem Unresolved	Propose	There is a slim possibility that this arc would be enabled if the pilot just wasn't thinking and sent the wrong primal message during the problem arc. There is virtually no way that between R1, OP, and RP the CDU would alter the primal message given by the pilot. This arc is supported by the same protocol nodes as in the problem arc.
Accept	Null	see normal feedback
	Neutral	see normal feedback
O Ack Abort	Neutral	The CDU acknowledges the recipient initiated abort by changing its state (i.e., the lights go off or the screen changes).
Acknowledge	Null	If the pilot takes the NULL instantiation in the Normal Feedback ARC, then most likely the CDU will use a NULL instantiation here.
	Neutral	The CDU may provide NEUTRAL feedback by indications of changes of state by: 1) showing an "on/off" light 2) changing a screen 3) highlighting a field within the screen but notice that none of these entities, separately, contain information about the Primary message!

Function Receiving Node

Arc	Form of Feedback	Annotations
E-Feedback	Inform	The CDU must supply feedback to the pilot about its current state regarding the communication components. This is accomplished through the information on the screens, once the particular button is pressed. For instance, when the COM key is depressed, the comm summary appears and displays the radios and associate modes and frequencies.
Primary	Null	Once the appropriate communication component screen is displayed and the pilot does nothing, this provides a strong belief to the CDU that the pilot is satisfied with that particular communication component. Until the pilot acts, the Feedback message is NULL.
	Inform	The pilot may send an overt message to the CDU that the pilot wants to see the communication components completed, i.e., 1) comm summary 2) data menu 3) initialisation menu 4) system status 5) test menu 6) secure communications menu 7) power management The pilot sends this message by editing the elements related to the communication components such as the radio, mode, frequency, etc.
Normal Feedback	Verify	The CDU transmits immediate feedback as the changes the pilot makes appears instantaneously on the screen. However, this does not mean that it is instantaneously active.

Edit	Inform	If the pilot has a strong belief of P1 but NOT P2, then the pilot has an opportunity to edit the message. That is the CDU has received the message but the message has not been adequately interpreted. In other words the pilot might have meant the COM key but pressed the NAV key by mistake. Yet this is highly unlikely because the NAV protocols are virtually identical with the COM protocols and the CDU has taken much care to distinguish the different lexicons. On the other hand, the similarity in protocols may cause a lessening of the belief leading to a less efficient transmission.
Accept	Verify	Any editorial changes made to the content of the message is immediately reflected in changes on the screens. The supporting protocols are as in normal feedback.
Acknowledge	Neutral	Acknowledge, Commit, and Abort all occur with the same keystroke. That is, it is ambiguous exactly what the pilot is acknowledging when the pilot hits the RETURN soft key or jumps to another communication component! This is not the best in design. In other cases (when pilot wants to move from one radio to another) the RETURN soft key has two functions: save the data, and go to another screen. This functions may want to be separated in a future design. Note that the message content is not saved until the pilot exits the screen!

Function Transmitting Node

Arc	Form of Feedback	Annotations
Primary	Inform	The E-feedback that the CDU receives is that the COM key has been struck. At this point the CDU sends a message to the pilot about the contents of comm summary. This is done by displaying the comm summary screen which summarises the radio and their respective mode and frequency.
Problem	Propose	The pilot has receive the message but does not have an adequate interpretation of the message and proceeds to propose the proper comm summary by changing radios, modes, and/or frequencies.
Finish	Neutral	The pilot has a strong belief of P1 and P2 at this point. That is the message has been received and properly interpreted. The pilot finishes the protocol by hitting return! N.B. this capability is not in the current setup. The only way a pilot can get out of comm summary is by pressing another function key. This implies that the CDU has NULL feedback that the pilot is satisfied with the radio link.
R Abort	Neutral	The pilot may choose to abort the Primary message by pressing another function key. N.B. that there is no explicit ABORT key.
Resolve	Inform	The CDU informs the pilot of the change in the Primary message by displaying the corrected information.
Problem Unresolved	Propose	The pilot is given the option of changing their mind. The same supporting protocol nodes are here as in the Problem arc.
P Go Direct	Neutral	The pilot has a strong belief of P1 and P2 at this point. That is the message has been received and properly interpreted. The pilot finishes the protocol by hitting return! N.B. this capability is not in the current setup. The only way a pilot can get out of comm summary is by pressing another function key. This implies that the CDU has NULL feedback that the pilot is satisfied with the radio link.
O Ack Abort	Neutral	The CDU must acknowledge an abort by changing screens.

Startup Receiving Node

Arc	Form of Feedback	Annotations
E-Feedback	Inform	The pilot must have some indication if the CDU is powered up or not. (e.g., a power on/off light or is it by default upon starting up the aircraft?)
Primary	Inform	The pilot must have a means of telling the CDU that the pilot wants it on/off.
Normal Feedback	Verify	Upon booting up, screen 0 appears. Upon shutting down, screen 0 disappears. The CDU provides verify feedback (similar to E-feedback).
Acknowledge	Neutral	There should be no need to provide an overt message acknowledging the reception of the message. However, in this case, the pilot acknowledges the "on" position by pressing the soft key called, INIT (see NULL).
	Null	There should be no need to provide an overt message acknowledging the reception of the message. However, in this case, the pilot acknowledges the "on" position by pressing the soft key called, INIT (see Neutral).

Startup Transmitting

Arc	Form of Feedback	Annotation
Primary	Null	Null is enable if the CDU is already in the desired state.
	Inform	"Display power on" "Present screen 0" So this needs a power on button and light and a screen.
Finish	Neutral	This is the most efficient GPG. The pilot has only two options: either the CDU is on and operating or it is not on nor operating. The pilot only needs to act at E-feedback and Problem to select/toggle their choice.
Problem	Propose	This is the most efficient GPG. The pilot has only two options: either the CDU is on and operating or it is not on nor operating. The pilot only needs to act at E-feedback and Problem to select/toggle their choice. A manual button that says, "power on/off" is required.
Resolve	Inform	The CDU must make visible the current state of the power and the operation of the unit. Upon startup, the CDU should initialise and test all systems (this can be an on going process in the background).
P Go Direct	Neutral	This is the most efficient GPG. The pilot has only two options: either the CDU is on and operating or it is not on nor operating. The pilot only needs to act at E-feedback and Problem to select/toggle their choice. same as finish.

Element Receiving Node

Arc	Form of Feedback	Annotation
E-Feedback	Inform	The pilot needs the state of the component displayed. The vis aud protocol supports this.
Primary	Null	The pilot may elect to send no message at this level since the element is in the desired state.
	Inform	The pilot informs the CDU that the pilot wants to change an element by interacting with the element. The motor protocol supports this message.
Normal Feedback	Null	The current CDU does not display ALL the elements related to a radio link. Therefore, there may be points in the conversation that both the pilot and the CDU must provide NULL feedback on the element that is not shown.
	Inform	If the CDU continually displays the elements then the type of feedback is inform. The disp soft and hard key protocol supports this message.
Acknowledge	Neutral	The pilot must acknowledge the state of the element. It is unclear how this is done in the current CDU interface design. There is confusion between this arc and abort.
Accept	Inform	It is unclear what the CDU does with the entry. Does it save the element? Does it activate the element? This arc is ambiguous.
Edit	Inform	see Primary.
Abort	Neutral	no comment.
Ack Abort	Neutral	This arc is deficient in the current design. The CDU does not confirm the pilot's wish to abort an entry by, for example, defaulting to the current state.

Element Transmitting Node

Arc	Form of Feedback	Annotation
Primary	Inform	The CDU informs the pilot about the elements by displaying them and linking them to the appropriate soft key. A good interface should have the inform instantiation all the time.
Finish	Neutral	In the current interface this arc is deficient, but there should be a way for the CDU of knowing that the pilot has adequately interpreted its message. This could be done in a proposed interface by selecting a completed link. The motor may be supporting protocol.
Problem	Inform	If the pilot can not interpret the fact that the CDU is trying to show an element (either because of incorrect content or wrong screen) then the pilot can send a message by pressing the appropriate hard and soft keys. The motor protocol will support this message.
Resolve	Inform	If for some reason the pilot has a problem with the display and activation of elements, the CDU must try again. The same nodes that support the Primary arc should support the Resolve Arc.
Problem Unresolved	Inform	If the pilot can not interpret the fact that the CDU is trying to show an element (either because of incorrect content or wrong screen) then the pilot can send a message by pressing the appropriate hard and soft keys. The motor protocol will support this message.
R Abort	Neutral	The element message is aborted if the pilot chooses to do so
O Ack Abort	Neutral	This arc is deficient in the current design. The CDU has no way of distinguishing P3 between the normal feedback route or through an unusual abort.
P Go Direct	Neutral	Like the finish arc, it is not clear if the CDU knows that the pilot has adequately interpreted its message.

AUD VIS Node

Arc	Form of Feedback	Annotation
E-Feedback	Inform	The E Feedback is the current visual state of the display.
Primary	Neutral	The pilot turns their attention towards the display. This is a neutral instantiation because there is no way of telling the CDU that the pilot is NOW looking in its direction, unless one had an eye tracker. The reason it is not quite NULL is because the CDU does not have a strong belief that the pilot is looking in its direction.
	Null	The pilot could imagine that the CDU takes for granted that the pilot will look at the CDU display. It is possible that the pilot may occasionally look away from the CDU and still interact manually.
Finish	Null	Regardless if the pilot is or is not looking at the CDU, the CDU will always believe that the pilot has looked at it.

Motor Node

Arc	Form of Feedback	Annotation
E-Feedback	Inform	The CDU provides a display of the buttons to be pressed (or the microphone to be spoken into).
Primary	Inform	The pilot touches (or speaks). The CDU must interpret the touch or the tone as a single message at this level.
Finish	Neutral	In the future one could have voice activation where the CDU repeats or displays the voice command.
	Null	Currently, the CDU has no force feedback to indicate to the pilot that a key has been touch. There may be some resistance to the press, but that's about all.

Disp Soft Node

Arc	Form of Feedback	Annotation
Primary	Inform	The CDU informs the pilot that the screens and associated soft keys are available to interact with, by displaying the screens and associated softkeys.
Finish	Neutral	The pilot may acknowledge the message by simply interacting with the screens and the softkeys.

Disp Hard Node

Arc	Form of Feedback	Annotation
Primary	Inform	The CDU provides the appropriate hard keys for the pilot to interact with.
Finish	Neutral	Neutral feedback is provided by the pilot touching the hard keys.

Appendix D

List of Interaction Deficiencies

List of Interaction Deficiencies

Although not apart of the Layered Protocol Analysis, the analyst may ask how the message is transmitted between pilot and CDU via the interface. If the interface does not have the capability of transmitting or receiving a particular message, then the interface is deficient with respect to that message. Concurrently, a new interface may be proposed that incorporates the missing Feedback messages and eliminates redundant messages where appropriate.

CDU protocol	*The CDU wants to see that a communication link is set*	
feedback form	*Current interface implementation*	*Proposed interface implementation*
E-FEEDBACK		
Inform	Both pilot and CDU know the CDU's capabilities.	The new interface may include a history of pilot-CDU interaction as well as a/c system data so that current recommendations can be made about the necessity of a communication link.
PRIMARY		
Null		
Inform	Function keys (grouped with other keys) and associated screens are displayed.	Function keys (separated from other keys) and associated screens are displayed.
NORMAL FEEDBACK		
Null		
Neutral	Pilot communicates with outside world which has no explicit relationship with the Primal message.	*same*
Verify	*not clear*	Press function key and select radio link or waypoint setting, etc.
PROBLEM		
Propose	Modify elements using soft and hard keys and navigating through several levels of menus	All modifiable elements related to the function are displayed concurrently on a single screen.
R ABORT		
Neutral	Select another function	*same*
RESOLVE		
Inform	COM Function key and COMM Summary screen	COM Function key and COMM Summary screen
ACCEPT		
Null		
Neutral	Screens and soft keys	Selecting a link or waypoint, etc. should automatically activate that link. Moving to another element should automatically save the changes done to the current element.
PROBLEM UNRESOLVED		
Propose	Screens and soft keys	*unlikely*
ACKNOWLEDGE		
Null		
Neutral	*deficient*	Highlight selected link
O ACK ABORT		
Neutral	*not clear*	Screen changes

feedback form	*Current interface implementation*	*Proposed interface implementation*
Receiving FUNCTION	*The Pilot wants to see all completed elements*	
E-FEEDBACK		
Inform	*deficient*	All editable elements are to be displayed simultaneously
PRIMARY		
Null		
Inform	appropriate fields and menus are available for the pilot to edit using the soft keys	appropriate fields and one nested menu are available for the pilot to edit using the rocker key
NORMAL FEEDBACK		
Verify	CDU highlights element being edited by changing screens and changing field background colour.	see E-FEEDBACK.
EDIT		
Inform	see PRIMARY.	see PRIMARY. The pop up menu assists the editing by listing all states including the current one.
ACCEPT		
Verify	see NORMAL FEEDBACK	see NORMAL FEEDBACK
ACKNOWLEDGE		
Neutral	*ambiguous*	clicking proposed ent saves element
ABORT		
Neutral	*deficient arc*	not clicking ent defaults to current state
ACK ABORT		
Neutral	*deficient arc*	current state is displayed

Transmit FUNCTION	*The CDU wants to see that the elements are displayed*	
E-Feedback		
Inform	Single button press	Memory of button presses
Primary		
Inform	COMM Summary screen	*same*
Finish		
Null		*remove instantiation*
Neutral	*deficient*	Highlighted link should be active link
Problem		
Propose	Pilot edits elements	*same*
R Abort		
Neutral	Move to another function	*same*
Resolve		
Inform	see PRIMARY	see PRIMARY
Problem Unresolve		
Propose	Modify elements	*same*
Go Direct		
Null		*remove instantiation*
Neutral	*deficient*	Highlighted link should be active link
O Ack Abort		
Neutral	*ambiguous*	default to current settings

feedback form	*Current interface implementation*	*Proposed interface implementation*
Receiving startup	*The Pilot wants to see all completed elements*	
feedback form	Current interface implementation	Proposed interface implementation
E-FEEDBACK		
Inform	Power button and light. (screen 0 is not continuously displayed)	Power button and light
PRIMARY		
Null		
Inform	Press Power button or start up aircraft	*same*
NORMAL FEEDBACK		
Verify	see E-FEEDBACK	see E-FEEDBACK.
ACKNOWLEDGE		
Neutral	Must press INIT key	*redundant*
Null	*not available*	*should be a background function*

Transmit startup	*The CDU want to see the power and startup states in the desired state*	
E-Feedback		
Inform	Its internal state and a button press	History of internal states and button presses
Primary		
Inform	"power on" button	green/red light on screen and perhaps soft key
Finish		
Null	*deficient*	only two options for this protocol - on/off. No feedback is required to the CDU confirming its power state.
Problem		
Propose	Toggle power	*same*
Resolve		
Inform	see PRIMARY	see PRIMARY. POW and TST functions should be background functions. An alert might be necessary when TST fails. This may require enabling an O abort arc.
Go Direct		
Null	*deficient*	see FINISH
Receiving ELEMENT	*The Pilot wants to see that the desired elements*	
E-FEEDBACK		
Inform	Header, Soft keys *redundant*	Matrix layout
PRIMARY		
Null		
Inform	Interact with softkeys	Interact with rocker, clr, and ent keys
NORMAL FEEDBACK		
Null	*may occur*	*should eliminate*
Verify	*ambiguous*	see E-FEEDBACK.
EDIT		
Inform	see PRIMARY	see PRIMARY
ACCEPT		
Verify	*ambiguous*	see NORMAL FEEDBACK
ACKNOWLEDGE		
Null	*deficient*	
Neutral	*ambiguous*	press ent
ABORT		
Neutral	*deficient*	Move to field without pressing ent
ACK ABORT		
Neutral	*deficient*	default to current state

feedback form	Current interface implementation	Proposed interface implementation
Transmit ELEMENT	*The CDU wants to see that the elements are displayed/active*	
E-Feedback Inform	Its internal state and a button press	History of states and button presses
Primary Inform	different softkeys and associated screen element	fields within matrix of COM screen
Finish Neutral	*deficient*	If the radio link elements are set as desired then selecting the link confirms P1 & P2.
Problem Propose	Select, Create, Edit, Save, Delete *not always intuitive*	Use rocker buttons to locate field. Use ent key to select. use rocker to navigate through menu items. Use numeric and clr keys to create, edit, and delete. Use ent key to save.
R Abort Neutral	select an alternate softkey	select an alternate field
Resolve Inform	display new state	display most states, highlight desired state, have an option to list all states (may invoking another screen), always a have a *new* field visible.
Problem Unresolve Propose	Modify elements	*same*
Go Direct Neutral	*deficient*	see Finish
O Ack Abort Neutral	*deficient*	default to current state
feedback form	*Current interface implementation*	*Proposed interface implementation*
Vis Aud protocol	*The Pilot wants to see the displays that contain the messages*	
E-FEEDBACK		
Inform	line of sight between CDU and eyes	*same*
PRIMARY Null Neutral	human looks	*same*
FINISH Null	CDU is passive	CDU is passive. An eye tracker could activate the CDU whenever the pilot is gazing at it. That would be Neutral

disp soft protocol	*The CDU wants to provide necessary software to transmit messages*	
E-FEEDBACK		
Inform	touch input	touch (or voice) input
PRIMARY Null Inform	screens and associated soft keys	fields and menu items (or audio)
FINISH Neutral	touch soft key	move cursor (or talk)

feedback form	*Current interface implementation*	*Proposed interface implementation*
motor protocol	*The Pilot wants to act upon the CDU by touch or voice*	
E-FEEDBACK		
Inform	location of CDU buttons	*same*
PRIMARY		
Null		
Inform	pilot touches	pilot touches (or speaks)
FINISH		
Null	CDU is passive	CDU is passive. An audio display may be used. That would be Neutral

hard key protocol	*The CDU wants to provide necessary hardware to transmit messages*	
E-FEEDBACK		
Inform	touch input	touch (or voice) input
PRIMARY		
Inform	associated hard keys	associated hard keys (or mic)
FINISH		
Neutral	touch hard key	touch hard key (or talk)

Appendix E

LPTool Limitations

LPTool Limitations

The following is a list of limitations that were noted during the development of the pilot-CDU interaction model. The items within the list were recorded chronologically as soon as the event occurred.

i) The Network View is two dimensional and becomes difficult to decipher (particularly which links are connected) once the number of protocols grow beyond ten. A three dimensional representation may solve this problem, in combination with colour codes.

ii) The protocol node labels must be capable of more than eight characters.

iii) LPTool does not allow the GPG to be cut and pasted in its entirety.

iv) The Network View landscape is limited.

The next points outline recommendations that would make the tool easier to use.

i) The speed with which windows are opened and closed is irritatingly too long.

ii) This version of the LPTool conflicts with some other Macintosh programs or inits causing the program to quit unexpectedly.

iii) The program sometimes refuses to allow the user to connect PNs. The connecting links will appear, however, the links will not be attached to the cursor, as they should be, or, they will not end where the cursor is.

iv) There are known complications with other Mac applications such as Kopy Kat and Ram Doubler.

v) Sometimes, the program saves files that are later unrecoverable. The tool will save in binary, a large file, and when the user attempts to open the file later, LPTool will tell the user the file is already open and a write protected, error #-49 occurred.

vi) The program will crash if the user places any special characters in the title i.e., punctuation and spaces.

Finally, the last items deal with problems that are not part of the programming.

i) The whole premise of the theory is dealing with mapping out interactions that are below the conscious level. The user, typically an engineer, might not pick up on the effectiveness of the tool because they are used to dealing with things that are concrete. Thus a team of philosophers might have an easier time.

ii) It is imperative that the designer acknowledges the fact that there are many means to the same end therefore, the D-model has the ability to take on many forms. Each D-model has the potential to become highly "personalised", therefore, documentation is imperative.

Recommendations

The next version of LPTool should attempt to incorporate the many changes that need to be taken care of in order to provide a medium to apply Layered Protocol Theory. Possible improvements could incorporate interactive models such as a Three Dimensional display that allows the user to rotate the diagram and scroll up and down it. The PN that the user is observing, should have the ability to highlight everything that is connected above and below it to allow easy reading of the interaction. Following the same theme, to facilitate easier reading of the D-model, diviplexed PNs should be able to be seen as one PN and then expanded if necessary. This is analogous to maintaining separate directories on a computer to allow quick distinction of all the main titles on disk. Another improvement that should be made deals with the usage of the tool when instantiating PNs. In this view, the tool essentially becomes a word processor, thus, the use of some word processor type functions could really be helpful. The user should also be able to view the entire D-model at once, so that the shape of the interaction can be observed.

The ultimate purpose of this tool is to give the user an efficient interface in which the tenets of LPT can be applied. When LPTool is revised, it is hypothesised that the application of LPT will become much easier, efficient and cost effective than conventional methods of engineering and analysis.

The LPTool program needs to be upgraded in order to perform a complete analysis including a simulation of the interaction. Ideally, one would apply an LPT analysis for an upgraded version of the LPTool program. One proposal is to rethink the program architectural philosophy based on a spreadsheet architecture. The analyst could edit or access the spreadsheet via a graphical user interface. This would significantly reduce the time to generated a full (uninstantiated) protocol node. Instead a protocol node would simply have graphical links to entries within the spreadsheet. Other recommendations include a three dimensional representation of the Network View model.

The spreadsheet architecture could accommodate the proposed dynamic simulation of belief states as Virtual messages are passed between communicative partners. The simulation should be capable of generating the corresponding protocol nodes and GPGs within the other partner, establish the proper Virtual message connections and follow a time evolution of the belief states within the GPG. The analyst would have to provide the simulation with initial belief states, statistical distribution of the message transmission times, and probabilities that the conversation would transverse a particular arc within the GPG.

The new version of the LPTool may be used to develop an ideal interaction model which could then be translated into design specifications for a new CDU interface. The next research task is to design a new interfaced based on

the results of the LPT analysis. The current interaction model may or may not be used as a basis for developing the protocols. An experimental study should be set up comparing the new and current CDU interfaces. This work hopes to take advantage of a new Aircraft Crewstation Demonstator being developed at DCIEM.

Finally, both Ecological Interface Design (EID) and Layered Protocol Theory assert that human-machine interfaces may improve with the application of these front end analysis techniques. EID asserts that a cognitively compatible interfaces begins with a complete description of the environment in which the system is designed to perform. The environment description is divided hierarchically. Ultimately, the information being shown by the interface must relate to all levels of the hierarchy so that the user may effectively carry out the task. Where EID concentrates on the form of the interface, LPT is primarily concerned with the necessary feedback links for effective communication. Further study is needed to determine the extent to which EID and LPT complement each other.

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Establishment sponsoring a contractor's report, or tasking agency, are entered in section 12.) Defence and Civil Institute of Environmental Medicine P.O. Box 2000, North York, Not. Canada M3M 3B9	2. DOCUMENT SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable) Unclassified

3. DOCUMENT TITLE (the complete document title as indicated on the title page. Its classification should be indicated be the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

Layered Protocol Analysis of a Control Display Unit (U)

4. DESCRIPTIVE NOTES (the category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DCIEM Report

5. AUTHOR(S) (Last name, first name, middle initial. If military, show rank, e.g. Burns, Maj. Frank E.)

Farrell, Philip S.E. and Semprie, Marc A.H.

6. DOCUMENT DATE (month and year of publication of document) November 1997	7.a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.) 59	7.b. NO. OF REFS. (total cited in document) 10

8.a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant) Not Applicable	8.b. CONTRACT NO. (if appropriate, the applicable number under which the document was written) Not Applicable
9.a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	9.b. OTHER DOCUMENT NO.(S) (any other numbers which may be assigned this document either by the originator or by the sponsor.) Not Applicable

10. DOCUMENT AVAILABILITY (any limitation on further dissemination of the document, other than those imposed by security classification)

- [√] Unlimited distribution
- [ ] Distribution limited to defence departments and defence contractors; further distribution only as approved
- [ ] Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
- [ ] Distribution limited to government departments and agencies; further distribution only as approved
- [ ] Distribution limited to defence departments; further distribution only as approved
- [ ] Other

11. ANNOUNCEMENT AVAILABILITY (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (10.) However, where further distribution (beyond the audience specified in 10) is possible, a wider announcement audience may be selected.)

12. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Department of National Defence
Mgen George R. Pearkes Bldg.
Ottawa, Ont CANADA K1A 0K2

DSIS DCD03
HFD 09/94

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

SECURITY CLASSIFICATION OF FORM
(Highest classification of Title, Abstract, Keywords)

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Layered Protocol Theory (LPT) has been described as a special case of Perceptual Control Theory (PCT) where its core tenet is, *All communication is the control of belief.* It was recognised that LPT could be used to analyse the interaction between communicating partners in the context of human-machine systems. System interface problems were identified for the Control Display Unit (CDU) in the CH-146 Griffon helicopter. This application presented a good opportunity to conduct a Layered Protocol analysis on the pilot-CDU system. Aspects of LPT were discussed in detail including the LPTool, its Network View, GPG View, and Nine Element View. A pilot-CDU interaction was modelled with the aid of the LPTool program. The analysis yielded a list of interaction deficiencies between pilot and CDU which supported previous observations. The deficiencies were addressed in a new interface design that would provide the necessary controls and displays so that the required messages could be successfully transmitted and interpreted.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible, keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Perceptual Control Theory
Layered Protocol Theory
Control Display Unit
Human Machine Systems
Human Machine Interaction
Feedback
Communication